UNIVERSITY OF CAMBRIDGE

High performance simulations of yield stress fluids in a structured adaptive mesh refinement framework with embedded boundaries

Knut Sverdrup

Supervisor: Dr. N. Nikiforakis

Cavendish Laboratory
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

Gonville and Caius College September 2019

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Knut Sverdrup
September 2019

Abstract

Viscoplastic fluids are a class of non-Newtonian liquids characterised by their yield stress. Unless an external stress is applied which is larger than this threshold value, the fluid does not flow, but exhibits rigid body behaviour. Above the yield stress, applied forces cause viscous deformation. Such fluids play important roles in a range of fields, notably in wellbore drilling, which is the application that motivated this project. One aspect of this operation requires displacement of drilling fluid by cement in the annulus between casing and geological surroundings, and both of these fluids are viscoplastics. Ensuring that this is done properly is of utmost importance to the overall safety of the drilling operation. Often, numerical simulations are the only viable way of experimenting with the effect of drilling parameters and fluid properties on the flow configuration and resulting behaviour. Unfortunately, the presence of a yield stress leads to a singularity in the apparent viscosity at zero strain. This causes substantial computational expense for the algorithms used to simulate fluid flow numerically, even when regularisation techniques are employed to alleviate the problem. Consequently, most published results in the literature on computational viscoplasticity has been restricted to two-dimensional and steady-state flows. In an attempt to address this, we have applied state-of-the-art techniques from high-performance computational fluid dynamics to the viscoplastic flow problem. Specifically, we utilise spatio-temporal adaptive mesh refinement on structured meshes in this context for the first time. This is achieved through the software framework AMReX, which includes state-of-the-art numerical tools for solving partial differential equations with optimal parallel scaling. The ability to rapidly simulate unsteady viscoplastic flow problems in three dimensions is demonstrated by novel numerical experiments in a lid-driven cavity. In order to investigate flows in more interesting domain geometries and around objects, an embedded boundary algorithm has been developed which works alongside the viscoplastic flow solver. We show how this methodology can be utilised to simulate flow inside non-rectangular objects, and investigate fully three-dimensional viscoplastic flow past several shapes of bodies for the first time.

Acknowledgements

I would like to acknowledge the EPSRC Centre for Doctoral Training in Computational Methods for Materials Science for funding under grant number EP/L015552/1. I also acknowledge the funding and technical support from BP through the BP International Centre for Advanced Materials (BP-ICAM) which made this research possible. Additionally, I am grateful to have received extra support towards living expenses through the Aker Scholarship.

Apart from funding, there are few things so essential to the successful completion of a Ph.D. as feedback, guidance and motivation from ones mentors. I am very grateful to Nikos Nikiforakis for inviting me to join the Cavendish Laboratory as his student. His ability to obtain industrial collaborations for academic research is remarkable, and I have benefited greatly from his substantial network of contacts and motivational conversations. Paulo Gomes has been a fantastic industrial mentor, with enthusiastic explanations of the intricacies of wellbore cementing operations and encouraging me to get in touch with all the best talent at Sunbury. Finally, Ann Almgren became an unexpected collaborator through my discovery and use of the AMReX software framework. Working together with such a highly talented mathematician, programmer and group leader has been a humbling experience, and I am especially grateful for the month I spent in her research group at the Berkeley Lab.

I have been very lucky to be working in the Centre for Scientific Computing, with an abundance of talented and helpful colleagues. Of special note is Arndt Ryo Koblitz, who has been instrumental to my understanding of the intricasies of yield stress fluids. Additional insight has come from fruitful discussions with Philip Blakely, Stephen Millmore, Louisa Michael, Nandan Gokhale and Simon Wilkinson. But the people who have really made the working environment special to me these years are the people I started alongside: Geraint Harcombe, Matthew Evans, Tomé Magalhães Gouveia, Brett Abram, William Grant, Benedetta Bianchi, Haran Jackson, Liam Pattinson and Mark Johnson.

Outside of the university, my wife and I have been accepted into a new culture with open arms, and made friends for life. Through Toby and Angela at the Blue Ball Inn, and subsequently Grantchester Cricket Club, we became part of a heartwarming and lively community during our years here. I would like to thank our new best friends: Michael and Abigail Elizabeth, James and Aimee, Philip and Olivia, Jake and Diana and Fergus and Oli. Thanks to you, we'll be back to visit England as often as we can.

Working towards this Ph.D. degree with such freedom has been a wonderful experience. At times, however, it has been difficult and rather isolating. I am deeply thankful for the constant support from my family at those times, because it made me realise that in the big picture, the problem I was stuck on was inconsequential.

Ingunn: thank you for being the sun of my life.

Table of contents

1 Introduction **1**

1.1 Motivation: cement displacement in wellbores 1

1.2 Novelty 9

1.3 Overview of the dissertation 11

2 Mathematical formulation **13**

2.1 Continuum fluid mechanics 14

2.2 Rheology 18

2.3 Governing equations 24

3 Numerical methodology **31**

3.1 Temporal discretisation 31

3.2 Hyperbolic fluxes 34

3.3 Projection methods for incompressible flow 37

3.4 Non-Newtonian fluids 40

4 High-performance computing **47**

4.1 Parallel processing and scalability 48

4.2 The AMReX framework 49

4.3 IAMR 55

4.4 Verification and validation 60

5 Large-scale simulations of yield stress fluids **69**

5.1 Extending IAMR to generalised Newtonians 70

5.2 Verification: Bingham plastic convergence study 75

5.3 Lid-driven cavity 79

5.4 Herschel-Bulkley fluids 87

5.5 Applicability of adaptive mesh refinement 90

5.6 Moving to three dimensions 91
5.7 Parallel performance 95

6 Immiscible displacement flow 97
6.1 Variable density: Rayleigh-Taylor instability 98
6.2 Volume of fluid method in IAMR 100
6.3 Plane channel displacement flow 103

7 Embedded boundaries for complex domain geometries 107
7.1 Revisiting the incompressible flow solver 109
7.2 Embedded boundary treatment 114
7.3 Validation studies 121
7.4 Objects translating through Bingham media 125

8 Conclusions 141
8.1 Proposed further work 143

References 147

Chapter 1

Introduction

This dissertation constitutes a summary of the work undertaken as part of my Ph.D. project throughout the last three years. As such, it is a relatively multifaceted piece of work, which is to be expected when a single document is required to capture the final state of what has been a continuous evolution. We will therefore start by taking a look at the original motivation for the project, so that the work presented in subsequent chapters fits in with an underlying narrative. Additionally, this will explain how our work towards a versatile software tool could be broken down into manageable pieces with clear milestones along the way. We will then go on to explain the novelties in our work, and list the peer-reviewed publications these findings have led to. Finally, we will give a brief overview of the contents in the remainder of the dissertation.

1.1 Motivation: cement displacement in wellbores

This Ph.D. project has been a collaboration with BP-ICAM, and its purpose was to make significant progress towards the development of a Computational Fluid Dynamics (CFD) tool to accurately simulate a range of fluid displacements for a cement job in real wellbore conditions. Cementing is an essential operation when drilling a well, and as such it is vital that our understanding of the underlying physical processes is strong. The resulting CFD tool should take into account rheology, flow rates, detailed wellbore and casing geometries, pipe movement, pressure and temperature. In addition to presenting a literature review which describes the problem from an engineering point of view, this section focuses on how much has already been achieved in published literature, and identifying which improvements can be made on existing endeavours.

This research originates in the importance of cementing operations when drilling production and injector wells. Failure to complete these jobs properly can potentially result in safety being compromised. Two of the processes which must be conducted in an optimal manner during cementing are mud conditioning, which requires treatment of the drilling fluid while it is being circulated in the system; and cement displacement, where the drilling fluid in the annular gap is replaced by cement. In order for on-site personnel responsible for these operations to make informed decisions, simulations can be performed in order to gain insight into the physical processes taking place and the resulting flow patterns downhole.

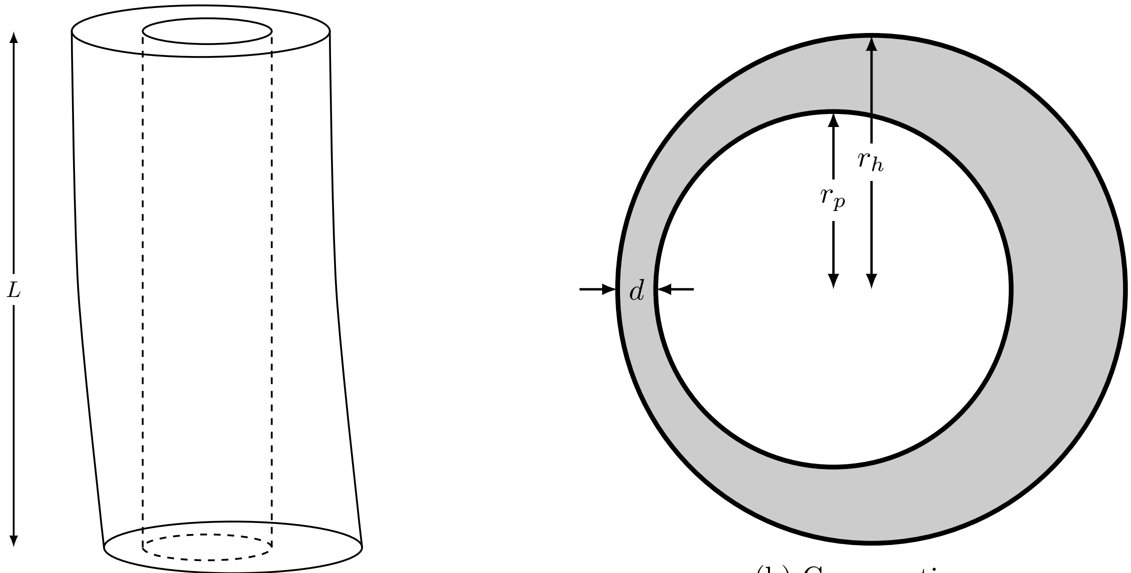
An engineering perspective

Wells in the oil and gas industry play the vital role of a safe pathway along which hydrocarbons can be extracted from the porous rock reservoir to the surface. In the construction of such wells, cementing is of paramount importance. The overall goal of the cementing operation is zonal isolation, i.e. separating distinguished permeable zones[1–3]. At the beginning of the cementing process, a drill bit has removed rock from a cylindrical hole in the Earth's crust. In order to avoid that the well collapses while this process is taking place, drilling mud is simultaneously pumped into the wellbore through the drill bit. Subsequently, the drill bit is removed and a casing is run. While the casing is being placed, the mud has time to gel, i.e. stop flowing. As a consequence of this, the mud must be circulated, analysed and altered until it achieves a set of stable, desirable properties. Mud fills up the interior of the casing, but also the annular gap between the casing and the borehole wall. The borehole wall separates the mud and the surrounding geological formation. At this point, the well is ready for cement to be pumped into the casing. Cement is injected from ground level, and travels downhole until it reaches the casing shoe. The cement then makes its way back upwards in the outer annulus. It is crucial that while doing so, all the mud initially in the annulus is displaced. In an unsuccessful displacement job, the cement can channel up the annulus while leaving gaps of mud still present. Additional benefits of using cement in these operations are protection from casing corrosion and sealing of loss zones.

A number of factors affect the success of the cementing operation. In order to achieve an optimal scenario, it is first of all desirable that the casing in the borehole is as centred as possible. In the ideal case it is concentric, but in practice this is unachievable, and the eccentricity varies with depth. In engineering papers, the eccentricity is defined by the percentage standoff χ, defined as[1;3;4]

$$\chi = \frac{d}{r_h - r_p} \cdot 100\%. \tag{1.1}$$

Here, d is the width of the smallest gap between borehole and pipe casing, while r_h and r_p are the radii of the hole and pipe, respectively (see Figure 1.1). Casing centralizers are used to keep the pipe as centred as possible[2;3]. Note that in reality, the borehole cross-section is not generally circular.



(a) Variable eccentricity

(b) Cross-section

Figure 1.1: Geometry of the borehole in a cementing problem with variable eccentricity. If the standoff is less than unity, the flow regime quickly becomes lopsided, leading to an uneven distribution of cement. In the cross-sectional view, the standoff is 50% and $r_p = 0.7r_h$.

It is essential that the cement slurry is kept separated from the mud at all times, to avoid undesirable interactions between the two. To this end, plugs are sent ahead of and after the cement when travelling through the casing[2;4]. To avoid mixing in the annulus, a layer of so-called spacer fluids and flushes separates the cement and mud[1;2]. Spacers also enhance the removal of solidified mud. Since drilling fluids belong to a class of non-Newtonian fluids with yield stress, leaving them in a static state causes the material to lapse into a gelled structure. In order for flow to become possible again, a yield criterion must be overcome[2;4].

Gelled mud can be further minimised by rotation and reciprocation of the casing, although this is not always possible. The movement is beneficial because it prevents the drilling fluid to settle, and the azimuthal flow due to casing rotation also helps avoid channelling in the annulus. Cement displacement normally takes place in the laminar flow regime for both mud and cement, in order to ensure a flat interface between the displaced and out of reach regions[1;4]. The flushes, however, are most often aqueous Newtonian fluids in the turbulent region, since this yields most efficient hole cleaning. Illustrations of the important considerations for wellbore drilling are shown in Figure 1.2.

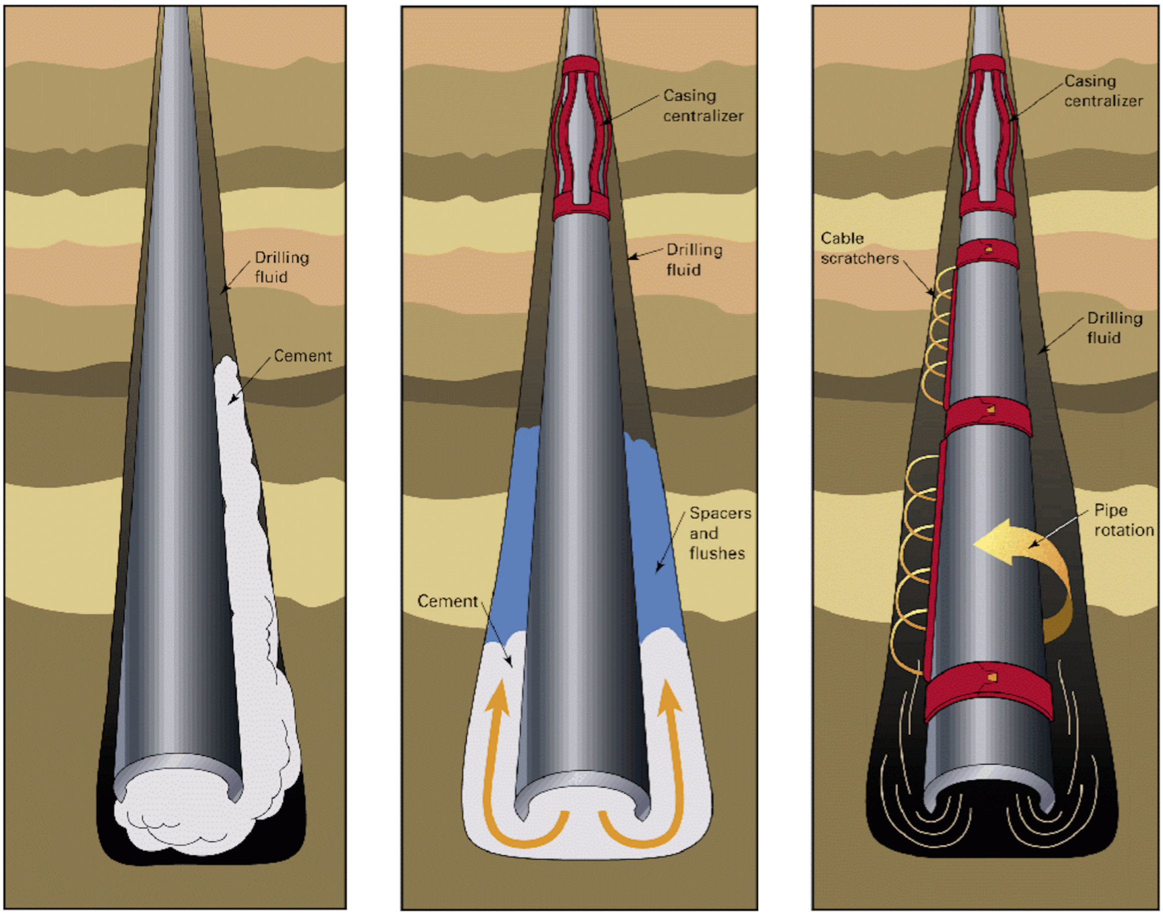


Figure 1.2: Left: Casing eccentricity leads to incomplete cementing jobs, so zonal isolation is not achieved. Middle: Spacers and flushes treat the drilling fluid and keep it separated from the incoming cement. Right: Movement of the casing distributes fluids evenly and helps overcome the yield criterion of the mud. Image courtesy of Crook, Benge, Faul and Jones[2].

Fluid properties and flow regime

There are primarily four important fluids involved in the mud displacement process: the drilling mud (or drilling fluid), which is a conditioned mixture of mud, water, oil, and cutting bits; the spacer, which separates the cement and mud and avoids undesirable interactions; the lead cement, which is pumped downhole first; and the tail cement, which completes the cementing operation, and which has a slightly higher density. Typical fluid properties used in industry are shown in Figure 1.3. Fluid rheology will be discussed at some depth in section 2.2, but we will point out a couple of noteworthy properties straight away. A corresponding rheological curve for a Newtonian fluid would be linear and intercept the origin. The cementing fluids deviate from this behaviour in two manners. Firstly, the curves intercept the secondary axis at a positive stress. The existence of such a yield stress is characteristic of viscoplastics, also known as yield-stress fluids. We shall be discussing these at some length, but also refer to a recent article which gives an excellent overview of their importance in the oil and gas industry[5]. Secondly, the relationship between stress and strain rate is highly nonlinear for stresses higher than the yield stress, hinting at so-called shear-thinning behaviour. Both of these properties are important to capture in our fluid models.

In addition to the rheology and characteristic fluid properties, it is important to have an idea about which flow regime the problem occurs in. As such, we seek approximate values of the Mach number (ratio of flow velocity to speed of sound), Reynolds number (ratio of inertial to viscous forces) and Bingham number (ratio of yield stress to viscous stress). The problem parameters used by important contributors to the literature are given in Table 1.1. Since we want to approximate the above dimensionless numbers using a single set of fluid properties, we take the average of displaced and displacing fluids for the dynamic viscosity, density and yield stress. The cross-sectional area of the annulus between the borehole and casing equals π times the difference between radii squared, so that with the given pump rate the average velocity through the pipe would be approximately 0.446 m/s and 0.159 m/s for the case of Savery et al.[6] and Enayatpour et al.[7], respectively. With the speeds of sound assumed to be at least hundreds of meters per second in these materials, it is safe to assume a flow with Mach number near zero (low-speed flow). Taking the difference between borehole and casing radii as the characteristic length, the Reynold's numbers for the two cases are approximately 593 and 18, respectively. These numbers are both well within the laminar flow regime, although the difference is quite large. According to the paper by Taghavi et al., relevant Reynold's numbers range between 200 and

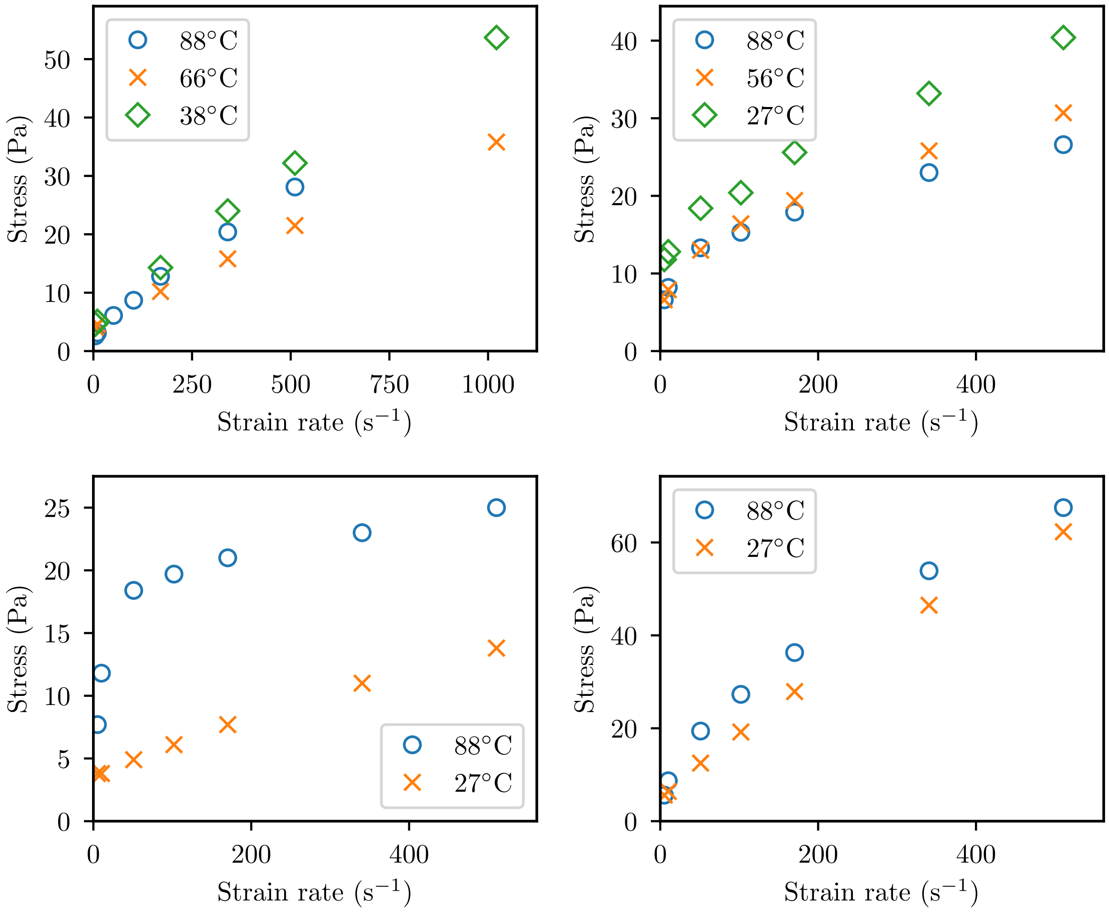


Figure 1.3: Relationship between stress and rate-of-strain for fluids present during cementing operations. The fluids depicted are drilling mud (top left), spacer (top right), lead cement (bottom left), and tail cement (bottom right).

1400[8]. We can compute approximate values of the Bingham number by noticing that the yield stress for the fluids in figure 1.3 is around 5-10 Pa. With the other parameters as discussed, this gives Bingham numbers of the order 10.

Current state of the art

Due to the importance of this system in drilling operations, a lot of effort has already been put forward to understand it. Theoretical advancements and experimental results have provided much insight, and in the last couple of decades computational approaches have received an increasing amount of attention. Early displacement models were

Table 1.1: Parameters used in previous articles on cement displacement simulators.

Variable	Savery et al.[6]	Enayatpour et al.[7]
Borehole diameter (m)	0.165	0.419
Casing diameter (m)	0.114	0.346
Pump rate (m^3/s)	0.005	0.007
Dynamic viscosity displaced fluid (Pa s)	0.02	0.60
Dynamic viscosity displacing fluid (Pa s)	0.03	0.55
Density displaced fluid (kg/m^3)	1198	1850
Density displacing fluid (kg/m^3)	1402	1800
Yield stress displaced fluid (Pa)	–	1.2
Yield stress displacing fluid (Pa)	–	1.4

all rule-based engineering approaches or based on reduced fluid models such as the Hele-Shaw cell[9–12].

The first comprehensive simulation of non-Newtonian flow through eccentric annuli was performed by Escudier, Oliveira and Pinho in 2002[13]. In this study, the incompressible Cauchy momentum equations were solved for a single fluid in cylindrical coordinates. The computational domain consisted of a two-dimensional collocated grid which represented a cross-section of the eccentric annulus, and no-slip boundary conditions were imposed. The inner wall (casing) was allowed to rotate. By considering the steady-state case, the authors were able to solve the equations by the SIMPLEC algorithm[14]. Several rheological equations for apparent viscosity were tested to model the non-Newtonian flow, among them those of power-law and regularised Herschel-Bulkley fluids[4].

Another milestone for mud displacement simulations in eccentric annuli was achieved when Savery, Darbe and Chin presented their methods for interface modelling in mud displacement simulations[6]. Unlike Escudier et al., they solved the governing equations equations in three dimensions, and allowed for both reciprocation and rotation of the casing wall. More important, however, was the inclusion of displacement of one fluid by another in the flow. By including a concentration function which described the relative amount of each of the two fluids, the apparent viscosity was included as an experimentally determined function of the fluid composition. Simplifications made in order to solve the system of equations include ignoring the radial momentum conservation and deeming the azimuthal pressure gradient negligible when compared to the drag force from casing rotation. The applicability of that code to prediction of residual cement was demonstrated in 2013 by Edwards et al.[15].

Very recent developments have also been made in the field of cement displacement simulators, most notably the works by Enayatpour and Oort[7] and Tardy et al.[16]. The former constitutes the most advanced simulator to date in terms of CFD capabilities. It solves the full incompressible Cauchy momentum equations in 3D for Herschel-Bulkley fluids. By tracking the interfaces between cementing fluids, a multiphase flow which models mud, spacer and cement is demonstrated. Tardy et al., on the other hand, demonstrate a solver with highly realistic wellbore geometries in deviated and horizontal wells. The simulation workflow consists of several distinct steps which are coupled to each other. First, a hydraulics simulation coupled with temperature is run. Subsequently, the centralisation of the string to be cemented is determined, before finally the results of these are fed as input into two displacement simulators (one for the interior of the casing; one for the annular gap on the exterior). The CFD solvers used in the last step are simplified by considering only 2D azimuthal axial formulations of the Navier-Stokes equations, and the mean values of velocity components are computed. According to the authors, this leads to only small inaccuracies, while reducing the computational runtime.

Oil and gas companies are becoming increasingly aware of the importance of computing accurate flow fields downhole during cement operations, but there is a gap between the needs of the industry and the current state of the art of cement displacement simulators. A need exists to develop new tools in order to reflect the complexity increase of the wells being drilled, and the intricate hardware deployed. Often the only option available is to use oversimplified models which are not accurate or precise enough to be applied in challenging well conditions, such as high pressure, high temperature, extended reach wells and wells with tight casing tolerances. More detailed simulations can be performed by commercial software, but, as a rule, they offer a generic platform for all kinds of flow which is not sufficiently tailored to the operational requirements of cementing and the timescales to perform usable simulations are far too long to meet operational time constraints.

In summary, current approaches to modelling primary cementing processes are rule-based systems, reduced models and CFD software. Every approach has its own advantages and limitations, and taking these into account, each has its own useful place in design and operations. For example, rule-based approaches aid the engineering design process, but given that they are phenomenological and usually rely on hydraulic analogy systems, they cannot provide a complete quantitative description of the process. The Hele-Shaw displacement model approach goes a long way to rectify such issues, but it is valid only when the fine-scale features of the displacement (close to the interface between the fluids) do not affect the bulk flow[10]. CFD software (given an appropriate underlying mathematical formulation) can provide more complete physical description of the problem, but it is not currently feasible to capture in detail the flow along the entire length of a borehole. Even for more limited studies, to run such models requires long computational time which is detrimental to applications on a regular basis. However, the considerable potential within this approach due to the continuing advancement of high-performance numerics and supercomputer architectures led us to base the project on pursuing it further. By working towards contributions which facilitate the simulation of yield stress fluids using contemporary high-performance computing, our efforts can aid the transition towards the next generation of cement simulations.

1.2 Novelty

Based on the aforementioned shortcomings, our goal has been to develop a software package which can capture rich fluid dynamics through efficient and massively scalable simulations of viscoplastic fluids. We wanted the ability to evolve time-dependent systems in three spatial dimensions, with the use of spatio-temporal adaptive mesh refinement in order to deal with the disparate length and time scales associated with the problem. Furthermore, we should be able to capture immiscible fluid-fluid displacement, in addition to flow in domains and around objects with non-trivial geometric features. Additional desirable features included variable-density flows, temperature dependency, and moving boundaries. In order to break down the goal into manageable chunks of work, we set ourselves intermediate goals which would be significant enough to warrant peer-reviewed publication, and try to achieve one of these at a time. This also meant that we automatically ensured that our work contributed to ongoing research in the academic field of computational fluid dynamics. We will presently list our novel contributions to the literature, but all the following will naturally be elaborated on throughout the remainder of the dissertation.

Apart from obtaining an overview of the current state of the art in the field through reviewing the available literature, the first and most comprehensive piece of work was the augmentation of a code called Incompressible Adaptive Mesh Refinement (IAMR) so that it could simulate non-Newtonian fluids rather than just Newtonian ones. IAMR employs a second-order accurate algorithm for solving the incompressible variable-density Navier-Stokes equations, and is based on the highly parallelisable code framework Adaptive Mesh Refinement for the eXa-scale (AMReX)[17], which has built-in support for structured adaptive mesh refinement, linear systems, and much more. By enhancing the code so that it can simulate the flow of any generalised Newtonian fluid, we have facilitated the first simulations of yield stress fluids within such a high-performance framework. A wide range of numerical experiments were conducted in the context of the lid-driven cavity. In doing so, we extended the available benchmark results for various fluid models. This included wider ranges of parameters for power-law and Bingham fluids, in addition to entirely new references for Herschel-Bulkley fluids. Additionally, reference results were provided for the first time for Bingham fluids in the three-dimensional case. The ability to simulate fully three-dimensional Bingham flow efficiently also allowed us to visualise the yield surface, both through temporal evolution and through steady-state results for various combinations of Reynolds and Bingham numbers. Finally, we demonstrated how structured AMR can be used to adaptively track the time-dependent yield surface[18].

The next piece of work was a relatively minor enhancement in comparison to the first, but important nevertheless. We implemented the volume-of-fluid method for the simultaneous simulation of two fluids, by tracking an advective scalar and allowing them to displace each other. In addition to validating our results in a well-documented Rayleigh-Taylor instability problem, we demonstrated that our code could handle channel displacement with a combination of Newtonian and viscoplastic fluids[19].

Finally, we published the first numerical methodology for simulating flow of generalised Newtonian fluids in complex domain geometries through the use of embedded boundary techniques. Much of the work was done while visiting the Center for Computational Sciences and Engineering at the Lawrence Berkeley National Laboratory. A lot of care was taken to ensure accurate computations of strain rate magnitude and apparent viscosity near the cut cells which represent solid boundaries, and we validated their implementation thoroughly. The new methodology enabled us to simulate a much wider range of interesting flow scenarios. In particular, we computed the first highly resolved 3D solutions of creeping flow around various objects translating at constant speed through a Bingham medium. In the case of flow around a sphere, we clarified some inconsistencies in the previously published literature[20].

Publications

Parts of this dissertation have been published in the following:

- K. Sverdrup, N. Nikiforakis and A. Almgren: "Highly parallelisable simulations of time-dependent viscoplastic fluid flow with structured adaptive mesh refinement." *Physics of Fluids* 30 (9) (2018) 093102.[18] (Named Editor's pick)
- K. Sverdrup, N. Nikiforakis and A. Almgren: "Numerical simulations of immiscible generalised Newtonian fluids" *Proceedings of ECCM VI and ECFD VII 11–15 June 2018, Glasgow, UK* (2019) 2969–2980.[19]
- K. Sverdrup, A. Almgren and N. Nikiforakis: "An embedded boundary approach for efficient simulations of viscoplastic fluids in three dimensions." *Physics of Fluids* 31 (9) (2019) 093102.[20] (Featured article, chosen for the cover of the monthly issue)

For all of the aforementioned articles, I have been the principal author. I carried out the numerical simulations, generated results and figures, and wrote the entire first drafts. My co-authors helped guide me with invaluable advice, and offered feedback and alterations for the manuscript. Additionally, Dr. Almgren was an expert troubleshooter and bugseeker during development.

1.3 Overview of the dissertation

In the following chapters, we will take a detailed look at the problems which we have solved, including their underlying formulation, the numerical tools employed and the physical insight obtained through numerical simulations. Chapter 2 provides a derivation and overview of the system of governing partial differential equations (PDEs) which describe the fluid dynamics, in addition to the rheological models which allow us to model non-Newtonian fluids through apparent viscosity functions. Chapter 3 reviews the most important numerical schemes utilised in our fluids solver, including temporal advancement, hyperbolic flux treatment, projection methods and regularisation of apparent viscosity functions containing singularities. In chapter 4, we introduce the AMReX framework in general and the IAMR code in particular, and explain how they stand out among high-performance software suites. We start to discuss our novel contributions in chapter 5, which gives the details of our paper on high-performance simulations of yield stress fluids in a three-dimensional cavity[18]. The extension to two-fluid displacement flows is given in chapter 6[19], while chapter 7 explains the addition of embedded boundaries in order to simulate flow in and around non-rectangular domain geometries[20]. Finally, chapter 8 offers some overall discussions and possibilities for further development, before the dissertation is concluded.

Chapter 2

Mathematical formulation

In order to properly explain the mathematical formulations used to describe the fluid mechanics, it is essential that we first introduce the mathematical notation and physical variables encountered. Our domain $\Omega \subset \mathbb{R}^d$ m is either in two or three domains, i.e. $d \in \{2, 3\}$, and has a boundary denoted by $\partial\Omega$. For vectors and rank-2 tensors, we prescribe the Euclidean scalar product and Frobenius inner product, respectively:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^\top \boldsymbol{b} \qquad = \sum a_i b_i \qquad \forall (\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{R}^d, \tag{2.1}$$

$$\boldsymbol{A} : \boldsymbol{B} = \mathrm{tr}(\boldsymbol{A}\boldsymbol{B}^\top) \qquad = \sum A_{ij} B_{ij} \qquad \forall (\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}^{d \times d}. \tag{2.2}$$

The induced norm of vectors is as usual $|\boldsymbol{a}| = \sqrt{\boldsymbol{a} \cdot \boldsymbol{a}}$, but we prescribe the scaled Frobenius norm

$$\|\boldsymbol{A}\| = \sqrt{\frac{1}{2}\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^\top)} \tag{2.3}$$

for rank-2 tensors, as is customary for convenience in viscoplastic fluid mechanics. On the other hand, the outer product of two vectors is written $\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^\top$, while component-wise multiplication (the Hadamard product) uses the notation $\boldsymbol{a} \odot \boldsymbol{b}$. We take the gradient of a vector $\boldsymbol{u}$ as the tensor with components

$$(\nabla \boldsymbol{a})_{ij} = \frac{\partial a_j}{\partial x_i}, \tag{2.4}$$

while the divergence of a tensor field is defined such that

$$(\nabla \cdot \boldsymbol{A})_j = \sum_{i=1}^{d} \frac{\partial A_{ij}}{\partial x_i}. \tag{2.5}$$

We also employ the notation

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \nabla \tag{2.6}$$

for material derivatives.

The physical variables which we encounter are all given in SI units. When dimensionless variables are introduced, a hat (e.g. $\hat{\boldsymbol{x}}$) will appear above them. The unit vector in the Cartesian x-direction is abbreviated $\hat{\boldsymbol{e}}_x$, with similar notation for other coordinates (Cartesian or otherwise). Variables are functions of position $\boldsymbol{x} \in \Omega$ and time $t \in \mathbb{R}\,\mathrm{s} \geq 0$. We denote by $\rho \in \mathbb{R}\,\mathrm{kg\,m^{-3}}$ the material density. The velocity field is introduced as $\boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^d\,\mathrm{m/s}$, with Cartesian components u, v and w. The Cauchy stress tensor $\boldsymbol{\sigma}(\boldsymbol{x}, t)$ is defined as the sum of isotropic and deviatoric parts, $\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} \in \mathbb{R}^{d\times d}\,\mathrm{Pa}$. Here, the pressure $p(\boldsymbol{x}, t) \in \mathbb{R}\,\mathrm{Pa}$ is multiplied by the identity tensor, while the deviatoric part of the stress tensor is denoted $\boldsymbol{\tau}(\boldsymbol{x}, t) \in \mathbb{R}^{d\times d}_{\mathrm{sym}}\,\mathrm{Pa}$. Finally, we introduce (twice) the symmetric part of the velocity gradient as the rate-of-strain tensor

$$\dot{\boldsymbol{\gamma}}(\boldsymbol{x}, t) = \nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\top \in \mathbb{R}^{d\times d}_{\mathrm{sym}}\,\mathrm{s}^{-1}. \tag{2.7}$$

2.1 Continuum fluid mechanics

Matter is made of atoms and molecules, which are discrete particles separated by space and interacting with each other in a manner depending on which of the four fundamental states (solid, liquid, gas or plasma) they are in. The physical theory best suited to describe the predominant physical behaviour of the matter depends on the length-scales at which it is observed. At inter-atomic distances, for example, quantum effects are highly relevant, but they are negligible for much larger length scales. At such large scales, it is reasonable to assume that the matter fills all space and is distributed continuously. The resulting theory is continuum mechanics, and has its origins in the pioneering works of Cauchy[21]. We shall be focusing on the case of fluids, i.e. gases and liquids, and particularly the latter.

Throughout this dissertation, the PDEs which we discuss are always variations of equations which describe conservation of mass and momentum. Starting from a general viewpoint, conservation of a physical variable ϕ in a fluid control volume Ω is governed by a continuity equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \phi\,\mathrm{d}V = -\int_{\partial\Omega} \phi\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S + \int_\Omega s\,\mathrm{d}V, \tag{2.8}$$

where $\boldsymbol{u}$ is the flow velocity, $\boldsymbol{n}$ is the unit normal vector pointing out of the control volume, and s denotes sources and sinks in Ω. It simply states that the temporal rate of change in the given quantity is the difference in the amounts flowing into and out of the domain, plus whatever is generated or lost due to the presence of source and sink terms. Utilising the divergence theorem and the Reynolds transport theorem[22] allows us to rewrite (2.8) as

$$\int_\Omega \frac{\partial \phi}{\partial t}\,\mathrm{d}V = -\int_\Omega \nabla \cdot (\phi \boldsymbol{u})\,\mathrm{d}V + \int_\Omega s\,\mathrm{d}V. \tag{2.9}$$

Note that this also holds for vector quantities, but that the product with $\boldsymbol{u}$ is then a dyadic outer product resulting in a rank-2 tensor. The equation can only hold for arbitrary control volumes if the integrand is zero, i.e.

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \boldsymbol{u}) = s. \tag{2.10}$$

Equation (2.10) serves as a very useful differential form of the general continuity equation. Mass conservation is straightforward to write down by taking $\phi = \rho$ and $s = 0$, since there are no sources or sinks for mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{2.11}$$

All non-relativistic momentum balance equations in the field of continuum mechanics can be derived as special cases of the general Cauchy momentum equation. The conserved quantity is the momentum density $\rho\boldsymbol{u}$, while the vector source term is decomposed as $\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f}$, where we have introduced $\boldsymbol{f}$ to describe external body forces such as gravity acting on the fluid. Recalling that $\boldsymbol{\sigma}$ is a sum of isotropic and deviatoric parts, we thus write down Cauchy's momentum balance in conservation form as

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}. \tag{2.12}$$

By specifying how the pressure p and deviatoric stress tensor $\boldsymbol{\tau}$ depend on other flow variables, specific applications ranging from solid mechanics to inviscid gas dynamics can be derived. For example, if we consider the case without any deviatoric stresses ($\boldsymbol{\tau} = 0$), (2.12) reduces to the Euler momentum equation[23], which governs adiabatic and inviscid fluid flow. It is part of the hyperbolic first-order system of PDEs known as the Euler equations, which are extensively utilised in simulations of ideal fluids when viscous effects are negligible[24–31].

Convective differential form

For numerical treatment, it can be useful to write the temporal derivative in (2.12) as a derivative of the velocity alone. This is possible by applying the chain rule to the first term and using (2.11) to remove the explicit dependency on the rate of change in density:

$$\frac{\partial(\rho\boldsymbol{u})}{\partial t} = \rho\frac{\partial\boldsymbol{u}}{\partial t} + \boldsymbol{u}\frac{\partial\rho}{\partial t} = \rho\frac{\partial\boldsymbol{u}}{\partial t} - \boldsymbol{u}\nabla\cdot(\rho\boldsymbol{u}) \tag{2.13}$$

Similarly, application of the chain rule to the second term in (2.12) gives

$$\nabla\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u}) = \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \boldsymbol{u}\nabla\cdot(\rho\boldsymbol{u}). \tag{2.14}$$

Adding these two equations together results in the convective differential form of the Cauchy momentum equation:

$$\rho\left(\frac{\partial\boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \nabla\cdot\boldsymbol{\tau} + \boldsymbol{f}. \tag{2.15}$$

Recalling the definition of the material derivative in (2.6), this form also makes it clear that the Cauchy momentum balance is merely an application of Newton's second law of motion. The left hand side is mass multiplied by material acceleration due to temporal change and convection, while the right hand side accounts for applied forces made up of hydrostatic effects, deviatoric stress divergence and external body forces.

Navier-Stokes equations

In fluid mechanics, few equations are as well known as Navier-Stokes. They govern the motion of simple viscous fluids, and have applications in aerodynamics[32–34], medicinal research[35;36] and petroleum engineering[37–40], to name a few. Named after Claude Navier and George Stokes for their major contributions[41;42] to their formulation in the first half of the nineteenth century, the equations have constituted a major field of research in their own right since their formulation, and continue to do so today. In fact, (dis-)proving the existence and uniqueness of their solutions is one of the seven Millennium prize problems[43].

The Navier-Stokes equations are an extension of (2.11) and (2.12) derived under a certain set of assumptions. Firstly, the Cauchy stress tensor must be Galilean invariant, which means that it can not depend directly on the velocity field, only on its gradient $\nabla \boldsymbol{u}$. Secondly, this dependency on the velocity gradient must be linear. Finally, the fluid must be isotropic. This means that the stress tensor is symmetric, so that it can be expressed as

$$\boldsymbol{\tau} = \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + \mu\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\top}\right), \tag{2.16}$$

where we have introduced the dynamic viscosity coefficient μ and the so-called second viscosity coefficient λ. These coefficients are fluid parameters, which generally depend on the state of the fluid. As will become clear shortly, the second viscosity coefficient only matters for compressible flows, and even then it is common to assume that Stokes' hypothesis, $\lambda = -\frac{2}{3}\,\mu$, holds[44]. Taking the divergence of (2.16) and inserting it into (2.15) gives the compressible Navier-Stokes momentum equation,

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) = -\nabla p + \lambda \nabla(\nabla \cdot \boldsymbol{u}) + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{f}. \tag{2.17}$$

Although the Navier-Stokes equations describe very generally the motion of many viscous fluids, the assumption of linearity in $\nabla \boldsymbol{u}$ and constant viscosity coefficients are too restrictive to model the non-Newtonian fluids typically encountered in wellbore cementing jobs. We can, however, safely assume incompressibility, since the flows are characterised by very low Mach numbers.

Incompressible flow

When the material density is constant within a control volume of the fluid, which moves with the fluid velocity, we call the flow incompressible. A common misconception is that incompressibility of flow implicates a constant density distribution. On the contrary many fluids exhibit a spatially varying density although it is constant within a fluid parcel. A vast amount of cases in continuum mechanics relate to incompressible flow, and several simplifications arise in the description of the fluid. By the definition of incompressibility, the material derivative of the density vanishes, i.e.

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0, \tag{2.18}$$

so that (2.11) is equivalent to the incompressibility constraint

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2.19}$$

Utilising this constraint in the momentum balance equation will lead to a simplified PDE, which is nonetheless accurate in lots of flow scenarios. For example, inserting it into (2.17) causes the term involving bulk viscosity to disappear, so that we are left with the incompressible Navier-Stokes momentum equation,

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{f}. \tag{2.20}$$

In addition to simplifying the mass and momentum balance equations, the assumption of incompressibility closes the system of equations since (2.19) actually corresponds to a very simple energy conservation equation, namely constant specific internal energy along streamlines of the flow. This can be traced to the fact that the pressure only has a thermodynamic meaning for compressible flows. Consequently, a separate conservation equation for energy is necessary in order to describe such flows.

2.2 Rheology

An applied shear strain acting on a fluid will cause flow through viscous deformation, and the field of rheology seeks to quantify the relationships between the applied strain (and its rate) and the induced shear stress resulting from it. Several mathematical models exist which describe the rheology of materials[45], but the most common method by far is to use rheological equations. These constitutive equations give the relationship between the stress and strain tensors and their temporal derivatives, in addition to physical variables such as temperature, pressure and species concentration. In their most general form they can contain quite complex dependencies. The rheological equation is a constitutive equation of fundamental importance for two reasons. Firstly, it allows us to treat the materials in question from a continuum point of view. It is thus the connection between the physics taking place on a microscopic scale and the continuum mechanics which we apply macroscopically. Secondly, it provides the missing equation to close the governing system of PDEs used to describe fluid flow. Without it, the system cannot be mathematically determined. Although there are exotic rheological phenomena such as thixotropy and rheopecty[46;47], we shall in the following restrict ourselves to relatively simple equations of state on the form $\boldsymbol{\tau} = \boldsymbol{\tau}(\|\dot{\boldsymbol{\gamma}}\|)$. In other words, the stress response is solely dependent on the magnitude of the rate-of-strain tensor.

Newtonian and non-Newtonian flow

Despite the fact that very few fluids are actually so-called Newtonian, most modelling of fluid mechanics is based on precisely that assumption. This is due to the major simplifications which occur for such a fluid, and the fact that some of the fluids which are most frequently modelled (notably water and air) are approximated well by the Newtonian constitutive equation. As a consequence, the term non-Newtonian is used to describe any type of flow which deviates from this simplest of examples.

Newtonian flow is characterised by a dynamic coefficient of viscosity $\mu > 0$ which is independent of the shear, yielding a linear relationship between rate-of-strain and stress in the rheological equation:

$$\boldsymbol{\tau} = \mu \dot{\boldsymbol{\gamma}}. \tag{2.21}$$

For fluids where the dependency of the stress on the rate-of-strain tensor is nonlinear, the apparent viscosity is a useful concept when considering rheological responses to shear. It is a generalisation of the constant viscosity for Newtonian flow, where we allow the viscosity to be a function of the magnitude of the rate-of-strain tensor. Denoting the apparent viscosity by η, we thus have

$$\boldsymbol{\tau} = \eta\left(\|\dot{\boldsymbol{\gamma}}\|\right) \dot{\boldsymbol{\gamma}}. \tag{2.22}$$

Many fluids are accurately modelled by a non-Newtonian behaviour that captures shear-dependency through a smooth increase or decrease in apparent viscosity. Such fluids include pseudoplastics (shear-thinning, $\partial\eta/\partial\|\dot{\boldsymbol{\gamma}}\| < 0$) and dilatants (shear-thickening, $\partial\eta/\partial\|\dot{\boldsymbol{\gamma}}\| > 0$). A model which captures this behaviour is the power-law fluid, with rheological equation

$$\boldsymbol{\tau} = \kappa \|\dot{\boldsymbol{\gamma}}\|^{n-1} \dot{\boldsymbol{\gamma}}, \tag{2.23}$$

and apparent viscosity

$$\eta = \kappa \|\dot{\boldsymbol{\gamma}}\|^{n-1}. \tag{2.24}$$

A specific power-law fluid is characterised by its dimensionless flow behaviour index $n > 0$ and consistency κ, which has units $\mathrm{Pa\,s}^n$). From (2.23) it is immediately clear that the Newtonian case with $n = 1$ separates pseudoplastics ($n < 1$) from dilatants ($n > 1$).

Viscoplasticity

Power-law fluids capture a range of important nonlinear features, but fluids occurring in cementing applications exhibit another rheological phenomenon which cannot be captured by the power-law model. This phenomenon is called viscoplasticity.

Viscoplastic fluids are non-Newtonian fluids which are characterised by a minimum induced stress necessary for flow to occur. For this reason they are also commonly referred to as yield-stress fluids. When the imposed stress does not exceed the threshold value, the material is modelled as a rigid solid. In regions where the yield stress is exceeded, however, the material flows like a fluid. The ability of the material to support a stress under certain circumstances gives rise to phenomena such as non-flat surfaces at rest under gravity and the coexistence of yielded (flowing) and unyielded (rigid) regions within the fluid. The former can be demonstrated by distorting the surface of mayonnaise in a jar: gravity alone is not strong enough to surpass the yield stress, and the surface remains in its distorted state. Coexistence of yielded and unyielded fluid regions can be exemplified by toothpaste being squeezed out of a tube: there is a finite shear layer near the tube wall, whereas the majority of the fluid flows as an unyielded plug with constant velocity. In addition to being fundamentally interesting from the perspectives of rheology, fluid mechanics and mathematical modelling, yield stress fluids occur naturally and are paramount to the success of animals such as mudskippers[48] and snails[49]. Their importance in industries ranging from medicine[50–53] to oil and gas exploration[4;5;8] has led to extensive research contributions in the field.

On a sub-continuum scale, the properties of yield stress fluids can be attributed to the fact that they are colloids, which are fluids with microscopical particles dispersed throughout them without mixing. These particles can either be solid particles, in which case the colloid is called a suspension, or another fluid. In the latter case, the colloid is known as an emulsion. We previously mentioned mayonnaise, which is an stable emulsion widely encountered in modern kitchens. Note that while colloids can be viscoplastic fluids, they need not be – vinaigrette is another commonly encountered kitchen emulsion which can be accurately modelled as a Newtonian.

Although the first mathematical model for plasticity in fluid mechanics was due to Schwedoff (1900)[54], most consider Eugene Bingham the father of yield-stress fluids due to his famous rheological equation[55] for simple shear flow, first published in 1916. James Oldroyd's extension to three-dimensional flows from 1947[56] is now referred to as the Bingham model, and constitutes the simplest possible viscoplastic fluid model. It captures the characteristic yield stress, but behaves as a Newtonian fluid with constant viscosity in the yielded region. Although this simplicity means it is a crude approximation for many viscoplastic fluids, it is also responsible for the widespread popularity of the model. A more comprehensive model, which allows smooth, nonlinear dependencies of the stress on the strain rate in the yielded region, was introduced by Winslow Herschel and Ronald Bulkley in 1926[57]. The viscosity in the yielded region is equivalent to that of a power-law fluid, allowing shear-thinning (pseudoplastic) and shear-thickening (dilatant) behaviour. Although more complex fluid behaviour can be captured by considering e.g. viscoelasticity[58] or thixotropy[47], the relative simplicity of the Herschel-Bulkley model compared to its wide applicability to relevant generalised Newtonian fluids has led it to become very popular. Other relevant fluid models which are designed to capture the same fluid behaviour include those of Casson[59] and, more recently, de Souza Mendes and Dutra[60]. In addition to the development of this range of mathematical models, recent research has focused on providing experimental links between the controllable fluid properties of colloids and the parameters which appear in the models[61]. For further reading on developments in viscoplastic fluids, we refer the reader to the review papers by Barnes[62] and Balmforth et al.[63].

Viscoplastic fluids have a stress threshold $\tau_0 > 0$ (the yield stress), below which they do not flow. We note that it is possible to describe elastic deformation for materials which do not flow, but we shall be considering constitutive equations which assume a rigid body approximation, i.e. zero strain rate below the yield stress. Regions where the stress satisfies $\|\boldsymbol{\tau}\| < \tau_0$ are known as unyielded regions, and they are of interest, for several reasons. Firstly, their existence contradicts the statement that "everything flows", given a large enough time scale. Yield-stress fluids thus possess the unique quality of finite cessation times in the absence of sufficiently strong driving forces[64]. This has been widely debated within rheology: is strict viscoplasticity a physically realistic attribute, or merely a mathematically convenient model?[65] The question is important to keep in mind when working on mathematical and computational models for yield stress fluids, but we can say with certainty that genuinely viscoplastic models provide useful descriptions of many materials. Unyielded regions are also referred to as plug regions, and in the cementing community one talks of a transition to gelled mud. When the fluid ceases to flow. The interface which separates yielded and unyielded regions is called the yield surface. Since the yield surface separates two fundamentally states of matter (viscous fluid flow and rigid solid behaviour), locating it precisely is a valuable metric used to validate computed flow distributions in a domain of interest.

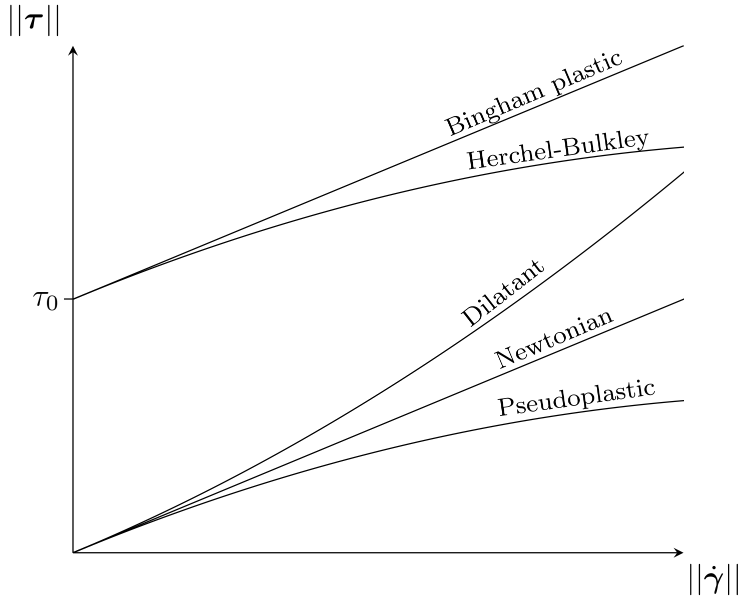


Figure 2.1: Classification of some generalised Newtonian fluids based on apparent viscosity.

As mentioned, the simplest type of viscoplastic fluid is the Bingham fluid[55;56], characterised by zero strain rate below the yield stress. In the yielded region, however, the stress depends linearly on the rate-of-strain magnitude, just like a Newtonian fluid. The stress-strain curve thus intercepts the $\|\boldsymbol{\tau}\|$-axis at the point $(0, \tau_0)$. As such, the Bingham fluid is a generalised Newtonian, with rheological equation

$$\begin{cases} \|\boldsymbol{\tau}\| \leq \tau_0 \,, & \text{if} \quad \|\dot{\boldsymbol{\gamma}}\| = 0 \\ \boldsymbol{\tau} = \left(\mu + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\|}\right) \dot{\boldsymbol{\gamma}} \,, & \text{if} \quad \|\dot{\boldsymbol{\gamma}}\| > 0 \end{cases} \tag{2.25}$$

and apparent viscosity

$$\eta = \mu + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\|}. \tag{2.26}$$

For Bingham plastics, we refer to μ as the plastic viscosity, and note that η has a singularity for $\dot{\boldsymbol{\gamma}} = 0$, as expected. Figure 2.1 illustrates how yield stress fluids are shifted on the stress-axis compared to power-law fluids.

Although the Bingham model captures the yield stress, which is the most characteristic property of the fluids we are trying to model, the Herschel-Bulkley fluid[57;66] facilitates an even better description of cement slurry and drilling fluids. It is a yield stress fluid with a nonlinear stress-strain dependency in the yielded region, and as such can be thought of as a hybrid between Bingham plastics and power-law fluids. The constitutive equation is

$$\begin{cases} \|\boldsymbol{\tau}\| \leq \tau_0\,, & \text{if} \quad \|\dot{\boldsymbol{\gamma}}\| = 0 \\ \boldsymbol{\tau} = \left(\kappa\,\|\dot{\boldsymbol{\gamma}}\|^{n-1} + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\|}\right)\dot{\boldsymbol{\gamma}} & \text{if} \quad \|\dot{\boldsymbol{\gamma}}\| > 0 \end{cases}, \tag{2.27}$$

while the apparent viscosity is

$$\eta = \kappa\,\|\dot{\boldsymbol{\gamma}}\|^{n-1} + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\|}. \tag{2.28}$$

Bingham plastics are a special case of Herschel-Bulkley fluids in exactly the same manner as Newtonian fluids follow a specific power-law constitutive equation. Plots of the rheological curves for Herschel-Bulkley fluids are shown in figure 2.2, for various values of n. Note that the Bingham fluid is recovered for $n = 1$.

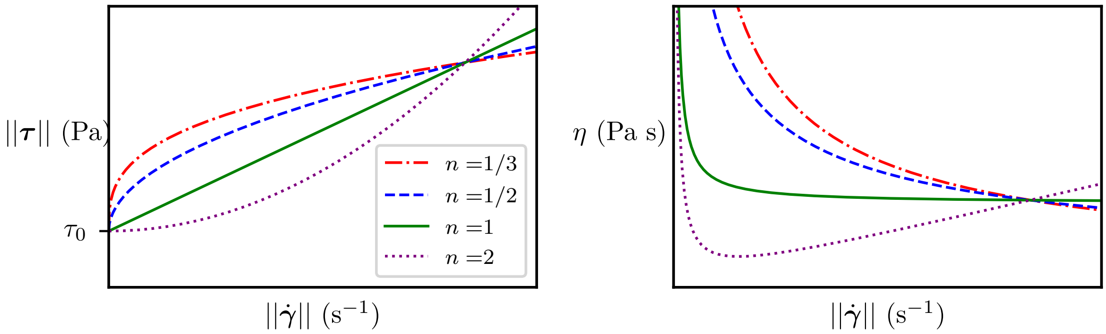


Figure 2.2: Rheological properties of Herschel-Bulkley fluids: stress magnitude (left) and apparent viscosity (right) as functions of the magnitude of the rate-of-strain tensor.

2.3 Governing equations

Our problem is thus, given a suitable domain, fluid parameters and initial conditions, to find $\rho(\boldsymbol{x},t)$, $\boldsymbol{u}(\boldsymbol{x},t)$ and $p(\boldsymbol{x},t)$ satisfying the following system of governing equations for $\boldsymbol{x} \in \Omega$:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0, \tag{2.29a}$$

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}, \tag{2.29b}$$

$$\boldsymbol{\tau} = \eta \left(\|\dot{\boldsymbol{\gamma}}\| \right) \dot{\boldsymbol{\gamma}}, \tag{2.29c}$$

$$\dot{\boldsymbol{\gamma}} = \nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\top}, \tag{2.29d}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.29e}$$

which must be completed by a generalised Newtonian apparent viscosity relation $\eta \left(\|\dot{\boldsymbol{\gamma}}\| \right)$ as described in section 2.2. Note that in cases when the density is constant, (2.29a) is obsolete. In addition to the above, we must have specified boundary values ρ_{BC} and $\boldsymbol{u}_{BC}$ on $\partial\Omega$ so that

$$\rho = \rho_{BC}\,, \quad \boldsymbol{u} = \boldsymbol{u}_{BC}\,, \quad \frac{\partial p}{\partial \boldsymbol{n}} = 0 \tag{2.29f}$$

there. The perfect no-slip boundary condition which we employ is the simplest and most common way to deal with wall effects for viscous flow, although many cases require more advanced treatments of the solid-fluid interactions, such as the stick-slip condition for Bingham fluids[67;68].

Equations (2.29) constitute a set of coupled nonlinear partial differential equations which are first order in time and second order in space, and which have both hyperbolic and parabolic terms.

Steady-state solutions

A special case of flow which is extensively used in mathematical and computational modelling of fluids is the steady state. It is characterised by a particle flow velocity which is constant in time, meaning that the material derivative is zero and the temporal and convective terms in (2.29b) disappear. Consequently, the flow is governed by the steady-state incompressible momentum equation

$$\nabla p - \nabla \cdot \boldsymbol{\tau} = \boldsymbol{f}. \tag{2.30}$$

Within numerical simulations of fluid motion, there are some algorithms for which the solution of the steady-state system as described by (2.30) is the most important step, even for transient solvers. This is possible because the steady-state system can be utilised to find the solution at a specific time, after which a temporal discretisation scheme can be coupled in to evolve the physical variables.

Nondimensionalisation

When considering specific problems or test cases, it is useful to use characteristic parameters of the problem to convert the physical variables and governing equations to dimensionless form. This minimises the amount of free variables necessary to describe the problem, and elucidates the dimensionless groups which govern the flow. If the problem has characteristic length $\mathcal{L}$ and speed $\mathcal{U}$, we typically take $\mathcal{L}/\mathcal{U}$ as the time scale and its reciprocal, $\mathcal{U}/\mathcal{L}$ as the characteristic strain-rate. Based on the apparent viscosity function, we also choose a characteristic viscosity $\mathcal{E}$, allowing us to scale stress and pressure by its product with the characteristic strain-rate, $\mathcal{E}\mathcal{U}/\mathcal{L}$. Appropriate conversions to dimensionless variables are then

$$\hat{\boldsymbol{x}} = \frac{1}{\mathcal{L}}\boldsymbol{x}, \quad \hat{\boldsymbol{u}} = \frac{1}{\mathcal{U}}\boldsymbol{u}, \quad \hat{t} = \frac{\mathcal{U}}{\mathcal{L}}t, \quad \hat{\dot{\boldsymbol{\gamma}}} = \frac{\mathcal{L}}{\mathcal{U}}\dot{\boldsymbol{\gamma}}, \quad \hat{\eta} = \frac{1}{\mathcal{E}}\eta, \quad \hat{p} = \frac{\mathcal{L}}{\mathcal{E}\mathcal{U}}p, \quad \hat{\boldsymbol{\tau}} = \frac{\mathcal{L}}{\mathcal{E}\mathcal{U}}\boldsymbol{\tau}. \tag{2.31}$$

Combination of the governing equations as given by (2.29) and the conversion relations yields the dimensionless governing equations

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\boldsymbol{u}} \cdot \hat{\nabla} \hat{\rho} = 0, \tag{2.32a}$$

$$Re\left(\frac{\partial \hat{\boldsymbol{u}}}{\partial \hat{t}} + \hat{\boldsymbol{u}} \cdot \hat{\nabla} \hat{\boldsymbol{u}}\right) = -\hat{\nabla}\hat{p} + \hat{\nabla} \cdot \hat{\boldsymbol{\tau}} + Ga\frac{\boldsymbol{f}}{|\boldsymbol{f}|}, \tag{2.32b}$$

$$\hat{\boldsymbol{\tau}} = \hat{\eta}\left(\left\|\hat{\dot{\boldsymbol{\gamma}}}\right\|\right)\hat{\dot{\boldsymbol{\gamma}}}, \tag{2.32c}$$

$$\hat{\dot{\boldsymbol{\gamma}}} = \hat{\nabla}\hat{\boldsymbol{u}} + \hat{\nabla}\hat{\boldsymbol{u}}^{\top}, \tag{2.32d}$$

$$\hat{\nabla} \cdot \hat{\boldsymbol{u}} = 0. \tag{2.32e}$$

Here, we have introduced the famous Reynolds number[22;69]

$$Re = \frac{\rho\mathcal{U}\mathcal{L}}{\mathcal{E}}, \tag{2.33}$$

which quantifies the relative effect of inertial forces to viscous ones, and the Galilei number

$$Ga = \frac{|\boldsymbol{f}|\mathcal{L}^2}{\mathcal{E}\mathcal{U}}, \tag{2.34}$$

which is so named since the external body force more often than not represents gravity, $\boldsymbol{f} = \rho \boldsymbol{g}$.

Additional dimensionless groups can arise from the scaling of the apparent viscosity function. For Newtonian fluids, we take $\mathcal{E} = \mu$ as the characteristic viscosity, so that the apparent viscosity function in dimensionless form is just

$$\hat{\eta}(\hat{\dot{\gamma}}) = 1. \tag{2.35}$$

Similarly, we let $\mathcal{E} = \kappa(\mathcal{U}/\mathcal{L})^{n-1}$ for power-law fluids, resulting in the apparent viscosity function

$$\hat{\eta}(\hat{\dot{\gamma}}) = \hat{\dot{\gamma}}^{n-1}. \tag{2.36}$$

For Bingham plastic fluids, we use the same characteristic viscosity as for Newtonian fluids, which results in a dimensionless apparent viscosity given by

$$\hat{\eta}(\hat{\dot{\gamma}}) = 1 + \frac{Bi}{\hat{\dot{\gamma}}}, \tag{2.37}$$

where the Bingham number is given by

$$Bi = \frac{\tau_0 \mathcal{L}}{\mu \mathcal{U}}. \tag{2.38}$$

The Bingham number quantifies the relative size of the yield stress to other viscous stresses. Finally, the generalisation to Herschel-Bulkley fluids is achieved by using $\mathcal{E} = \kappa(\mathcal{U}/\mathcal{L})^{n-1}$ again. This gives

$$\hat{\eta}(\hat{\dot{\gamma}}) = \hat{\dot{\gamma}}^{n-1} + \frac{HB}{\hat{\dot{\gamma}}} \tag{2.39}$$

for Herschel-Bulkley fluids, where we have introduced the Herschel-Bulkley number

$$HB = \frac{\tau_0 \mathcal{L}^n}{\kappa \mathcal{U}^n}. \tag{2.40}$$

An illustrative example

So far, we have described mathematically which equations govern the flow of a generalised viscoplastic fluid acted on by unprescribed forces in an arbitrary domain. In order to obtain a more intuitive understanding of viscoplastic behaviour, we turn to the specific example of Poiseuille flow for an incompressible Bingham fluid, for which an analytical solution exists at steady-state. The governing equations are (2.30) with $\boldsymbol{f} = 0$, completed by the Bingham constitutive law given in (2.25). The analytical solution was presented by Frigaard, Howison and Sobey in 1991[70], but their paper only gave the outline for the solution procedure and focused on stability analysis. We therefore include the full derivation here.

Consider a rectangular domain in two dimensions, which stretches infinitely in the x-direction and has width $2\mathcal{W}$ in the y-direction, with $y = 0$ symmetrically dividing the domain in half. The domain is filled with a Bingham fluid characterised by its plastic viscosity μ and yield stress τ_0. A constant pressure gradient $\mathcal{P}/\mathcal{W}$ is applied throughout the domain, which results in a steady flow with maximum velocity $\mathcal{U}$, to be determined. We take $\mathcal{W}$ and $\mathcal{U}$ as characteristic length and speed, respectively, and scale the apparent viscosity by μ. In dimensionless form, the governing equations are then the steady-state Cauchy momentum equation and the incompressibility condition,

$$\hat{\nabla} \cdot (-\hat{p}\boldsymbol{I} + \hat{\boldsymbol{\tau}}) = 0, \tag{2.41a}$$

$$\hat{\nabla} \cdot \hat{\boldsymbol{u}} = 0, \tag{2.41b}$$

completed by the Bingham constitutive law given by

$$\begin{cases} \hat{\dot{\gamma}} = 0 & \text{if} \quad \|\hat{\boldsymbol{\tau}}\| \le Bi, \\ \hat{\boldsymbol{\tau}} = \left(1 + \dfrac{Bi}{\|\hat{\dot{\gamma}}\|}\right)\hat{\dot{\gamma}} & \text{if} \quad \|\hat{\boldsymbol{\tau}}\| > Bi. \end{cases} \tag{2.42}$$

The pressure is given by $\hat{p} = -Po\,\hat{x}$, where the Poiseuille number

$$Po = \frac{\mathcal{P}\mathcal{W}}{\mu\mathcal{U}} \tag{2.43}$$

quantifies the relative strength of the pressure gradient. The flow is unidirectional, so the solution is given solely by $\hat{u}(\hat{y})$. Furthermore, since the only non-zero components of the rate-of-strain tensor are $\hat{\dot{\gamma}}_{12} = \hat{\dot{\gamma}}_{21} = \frac{\partial \hat{u}}{\partial \hat{y}}$, we know from (2.42) that $\hat{\boldsymbol{\tau}}$ also has diagonal elements equal to zero. In order to find the dependency of $\hat{\tau}_{12}$ on $\hat{y}$, we integrate (2.30) over a rectangle with lower left corner at $(-a,-b)$ and upper right corner at (a,b). First of all, we employ the divergence theorem:

$$\int_{\Omega} \hat{\nabla}\cdot\left(-\hat{p}\boldsymbol{I}+\hat{\boldsymbol{\tau}}\right)\mathrm{d}V = \oint_{\partial\Omega}\left(-\hat{p}\boldsymbol{I}+\hat{\boldsymbol{\tau}}\right)\cdot\boldsymbol{n}\mathrm{d}S, \tag{2.44}$$

where $\boldsymbol{n}$ is the outward-pointing normal vector on the domain boundary. We shall compute both of these integrals and equate them to zero. Starting with the right-hand side, we write out all the terms of the (counter-clockwise) line integral in full, and obtain

$$\begin{aligned}
\oint_{\partial\Omega}\left(-\hat{p}\boldsymbol{I}+\hat{\boldsymbol{\tau}}\right)\cdot\boldsymbol{n}\mathrm{d}S &= \oint_{\partial\Omega}\left(-\begin{bmatrix}\hat{p} & 0\\ 0 & \hat{p}\end{bmatrix}+\begin{bmatrix}0 & \hat{\tau}_{12}\\ \hat{\tau}_{12} & 0\end{bmatrix}\right)\cdot\boldsymbol{n}\mathrm{d}S\\
&= \int_{-a}^{a}\begin{bmatrix}Po\,\hat{x} & \hat{\tau}_{12}(-b)\\ \hat{\tau}_{12}(-b) & Po\,\hat{x}\end{bmatrix}\cdot\begin{bmatrix}0\\ -1\end{bmatrix}\mathrm{d}\hat{x}+\int_{-b}^{b}\begin{bmatrix}Po\,a & \hat{\tau}_{12}(\hat{y})\\ \hat{\tau}_{12}(\hat{y}) & Po\,a\end{bmatrix}\cdot\begin{bmatrix}1\\ 0\end{bmatrix}\mathrm{d}\hat{y}\\
&+\int_{a}^{-a}\begin{bmatrix}Po\,\hat{x} & \hat{\tau}_{12}(b)\\ \hat{\tau}_{12}(b) & Po\,\hat{x}\end{bmatrix}\cdot\begin{bmatrix}0\\ 1\end{bmatrix}\mathrm{d}\hat{x}+\int_{b}^{-b}\begin{bmatrix}-Po\,a & \hat{\tau}_{12}(\hat{y})\\ \hat{\tau}_{12}(\hat{y}) & -Po\,a\end{bmatrix}\cdot\begin{bmatrix}-1\\ 0\end{bmatrix}\mathrm{d}\hat{y}\\
&= \begin{bmatrix}-2a(\hat{\tau}_{12}(-b)+\hat{\tau}_{12}(b))\\ 2\int_{-b}^{b}\hat{\tau}_{12}(\hat{y})\mathrm{d}\hat{y}\end{bmatrix} = 0,
\end{aligned}\tag{2.45}$$

where we have taken advantage of the fact that $\boldsymbol{\tau}$ has no x-dependency. The important result from this is that τ_{12} is anti-symmetrical, so $\tau_{12}(-y) = -\tau_{12}(y)$. We will presently compute the integral again, but without employing the divergence theorem first. Taking the divergence of the stress tensor, we have

$$\hat{\nabla}\cdot\hat{\boldsymbol{\tau}} = \begin{bmatrix}\frac{\partial\hat{\tau}_{11}}{\partial\hat{x}}+\frac{\partial\hat{\tau}_{12}}{\partial\hat{y}}\\ \frac{\partial\hat{\tau}_{12}}{\partial\hat{x}}+\frac{\partial\hat{\tau}_{22}}{\partial\hat{y}}\end{bmatrix} = \begin{bmatrix}\frac{\partial\hat{\tau}_{12}}{\partial\hat{y}}\\ 0\end{bmatrix}. \tag{2.46}$$

Since the second component of (2.41a) is identically zero, we consider the first component only. The left-hand side integral in (2.44) is thus

$$\begin{aligned}
\int_{\Omega}\left(-\hat{\nabla}\hat{p}+\hat{\nabla}\cdot\hat{\boldsymbol{\tau}}\right)\mathrm{d}V &= \int_{\Omega}\left(Po+\frac{\partial\hat{\tau}_{12}}{\partial\hat{y}}\right)\mathrm{d}V = 4abPo+\int_{-a}^{a}\int_{-b}^{b}\frac{\partial\hat{\tau}_{12}}{\partial\hat{y}}\mathrm{d}\hat{y}\mathrm{d}\hat{x}\\
&= 4abPo+2a\left(\hat{\tau}_{12}(b)-\hat{\tau}_{12}(-b)\right) = 4a\left(bPo+\hat{\tau}_{12}(b)\right) = 0,
\end{aligned}\tag{2.47}$$

where we have used the anti-symmetrical nature of τ_{12}. Since b is an arbitrary distance from the centre of the channel, the off-diagonal components of the stress tensor are

$$\hat{\tau}_{12}(\hat{y}) = -Po\,\hat{y}. \tag{2.48}$$

At the walls, the dimensionless stress magnitude equals Po, and if this value exceeds Bi, then flow will occur in the channel. Yielded and unyielded regions are thus separated by the characteristic position $y_0 = Bi/Po = \tau_0/\mathcal{P}$. In the yielded region where $y_0 < \hat{y} \leq 1$, the constitutive equation can be written

$$\hat{\tau}_{12} = \hat{\dot{\gamma}}_{12} + Bi\,\frac{\hat{\dot{\gamma}}_{12}}{\left\|\hat{\dot{\gamma}}\right\|} = \frac{\partial \hat{u}}{\partial \hat{y}} + Bi \ \mathrm{sgn}\left(\frac{\partial \hat{u}}{\partial \hat{y}}\right). \tag{2.49}$$

Using (2.48) and differentiating, the first component of velocity satisfies

$$\frac{\partial^2 \hat{u}}{\partial \hat{y}^2} = -Po, \tag{2.50a}$$

$$\hat{u}\big|_{\hat{y}=1} = 0, \tag{2.50b}$$

$$\left.\frac{\partial \hat{u}}{\partial \hat{y}}\right|_{\hat{y}=y_0} = 0. \tag{2.50c}$$

By integrating (2.50) twice with respect to y, and using the boundary conditions to get rid of the constants of integration, we arrive at a velocity profile given by

$$\hat{u}(\hat{y}) = \frac{1}{2}(1-y_0)^2 Po\left(1 - \left(\frac{\hat{y}-y_0}{1-y_0}\right)^2\right), \tag{2.51}$$

where $\frac{1}{2}(1-y_0)^2 Po$ is the dimensionless velocity of the fluid in the limit $\hat{y} \to y_0$. Consequently, we can deduce that $\mathcal{U}$ must be chosen such that this limiting value equals unity, i.e.

$$\mathcal{U} = \frac{(1-y_0)^2}{2y_0}\frac{\tau_0\mathcal{W}}{\mu}. \tag{2.52}$$

The Bingham and Poiseuille numbers then reduce to

$$Bi = \frac{2y_0}{(1-y_0)^2} \tag{2.53}$$

$$Po = \frac{2}{(1-y_0)^2}, \tag{2.54}$$

and it is clear that the only free variable of the dimensionless system is y_0, the ratio of viscoplastic forces to those induced by the pressure gradient.

When we perform the same steps for the other yielded region ($-1 \leq \hat{y} < -y_0$), the only difference in the result is that $\hat{y} \to -\hat{y}$. Finally, we must consider the unyielded region in the middle of the domain ($|\hat{y}| \leq y_0$). According to (2.25), the velocity is constant. Since the velocity field must be continuous, this constant value is easily found by using the result for the unyielded region at the limit $\hat{y} = y_0$. The velocity profile which describes Poiseuille flow for a Bingham fluid is therefore

$$\hat{u}(\hat{y}) = \begin{cases} 1 & |\hat{y}| \leq y_0, \\ 1 - \left(\frac{|\hat{y}|-y_0}{1-y_0}\right)^2, & y_0 < |\hat{y}| \leq 1, \end{cases} \tag{2.55}$$

Figure 2.3 shows the velocity profile for various values of y_0. Finally, an important characteristic is the average velocity, given by

$$\bar{u} = \int_0^{y_0} \mathrm{d}\hat{y} + \int_{y_0}^1 \left(1 - \left(\frac{|\hat{y}| - y_0}{1 - y_0}\right)^2\right) \mathrm{d}\hat{y} = 1 - \int_{y_0}^1 \left(\frac{\hat{y} - y_0}{1 - y_0}\right)^2 \mathrm{d}\hat{y} \ = \left(\frac{2 + y_0}{3}\right). \tag{2.56}$$

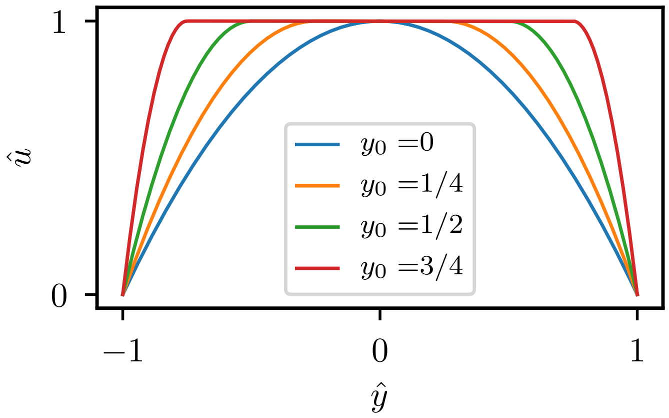


Figure 2.3: Velocity profile for a Bingham fluid flowing in a rectangular duct due to a constant pressure gradient. The value of the flow parameter y_0 is given in the legend. Note that for $\tau_0 = 0$ we recover the parabola which characterises Newtonian flow.

Chapter 3

Numerical methodology

Due to the applicability of the partial differential equations which govern fluid mechanics to a vast amount of scenarios, methods for computing their solution is one of the most studied problems in numerical analysis. Due to the generality of their most basic formulation, the numerical toolbox required to handle different challenges is large. As such, many different algorithmic approaches have been published in the literature, which deal with various aspects of the problem. In this chapter, we review some of the most important numerical tools utilised throughout our work, starting with the general approach to time-marching schemes and hyperbolic flux computation, and moving through to the more specific problems of enforcing incompressibility and dealing with singularities in the apparent viscosity functions for viscoplastics.

3.1 Temporal discretisation

For any physical system with flow which varies in time, it is natural to take a known starting point and consider the evolution of the system as a timer starts running. Similarly, when solving PDEs which are first order in time numerically, we seek algorithms which allow us to gradually march forward in time from an initial configuration, accurately predicting the evolution of the system as we go. Such algorithms are called time-marching schemes, and are generally facilitated by the Method Of Lines (MOL).

Consider a PDE for an unknown $\phi(\boldsymbol{x}, t)$ which is first order in time, and which otherwise can depend on position and time, in addition to ϕ itself and its spatial derivatives up to some order, which we take as two. Then it can be rewritten in the form

$$\frac{\partial \phi}{\partial t} = R(\phi, \boldsymbol{x}, t, \nabla\phi, \nabla^2\phi), \tag{3.1}$$

and our task is to design an algorithm which predicts $\phi(\boldsymbol{x}, t > 0)$. The idea behind the method of lines is to reduce the problem complexity by first converting the PDE to an ordinary differential equation (ODE) through computing discrete approximations to the spatial derivatives in all cells of the domain. This is typically done using finite difference, finite element or finite volume methods. We denote by a subscript index vector $\boldsymbol{i} = (i, j, k)$ the cell averages in the corresponding grid cell. Consequently, we can introduce the short hand notation

$$R_{\boldsymbol{i}}(t) = R\left(\phi_{\boldsymbol{i}}(t), x_{\boldsymbol{i}}, t, (\nabla\phi)_{\boldsymbol{i}}(t), (\nabla^2\phi)_{\boldsymbol{i}}(t)\right) \tag{3.2}$$

for the cell-averaged value in the right hand side of (3.1) as a function of time alone. This leaves an ODE to be solved numerically:

$$\frac{\partial\phi_{\boldsymbol{i}}}{\partial t} = R_{\boldsymbol{i}}(t). \tag{3.3}$$

Note that although (3.3) appears to have decoupled the inter-cell dependency, this is not true since the cell-averaged spatial discretisation generally depend on values in neighbouring cells. Even so, we don't need to consider estimates of these quantities in neighbours when doing the temporal update, and consequently can drop the subscript $\boldsymbol{i}$ for the remainder of this section.

Algorithms for initial value problems for ODEs

Having simplified the problem at hand, we now discretise time, so that our problem is to find the solution to (3.3) at time t^{m+1}, taking the system at time t^m as initial conditions. We denote the time step $\Delta t = t^{m+1} - t^m$ and use superscripts to identify the computed values at the corresponding times, i.e. $\phi^m = \phi(t^m)$. Integrating over the time step and evaluating spatial averages at the discrete points in time then leads to the update formula

$$\phi^{m+1} - \phi^m = \int_{t^m}^{t^{m+1}} R(t)\,\mathrm{d}t = \Delta t\left(\vartheta R^{m+1} + (1-\vartheta)R^m\right), \tag{3.4}$$

where ϑ is a parameter which determines the weighting of the old and new R-values in the integration. It is sometimes referred to as the implicitness parameter, because its value determines that important characteristic of the scheme. A scheme is explicit if $\vartheta = 0$, so that the update formula only depends on the computed values at the old time. If $\vartheta > 0$ so that there is any dependence on the new time values, the scheme is implicit. We call it fully implicit if $\vartheta = 1$. These two extreme cases are known respectively as the Forward and Backward Euler methods, and are both first-order accurate in time. The trapezoidal rule can be utilised in order to obtain the popular second-order accurate semi-implicit scheme known as the Crank-Nicholson method. Explicit methods are much simpler to use than implicit ones, but implicit methods have much better stability properties, meaning that larger time steps can be taken safely. Note however, that stability of a time-marching scheme does not guarantee high accuracy without taking the time step size into consideration.

Predictor-corrector schemes were developed in order to combine the simplicity of explicit schemes with the stability and robustness associated with implicit ones. An intermediate value $\tilde{\phi}^{m+1}$ is predicted using the forward Euler scheme, before this intermediate value is used to correct the approximation with a Crank-Nicholson update[71]:

$$\tilde{\phi}^{m+1} = \phi^m + \Delta t R^m, \tag{3.5a}$$

$$\phi^{m+1} = \phi^m + \frac{1}{2}\Delta t(R^m + \tilde{R}^{m+1}). \tag{3.5b}$$

Here, we use the notation $\tilde{R}^{m+1}$ to describe an evaluation of R using $\tilde{\phi}^{m+1}$. Note that such a predictor-corrector scheme is very similar to the simplest Runge-Kutta type algorithm[72;73], which uses the forward Euler scheme to predict values at the intermediate time $t^{m+1/2}$ and subsequently applies the midpoint rule:

$$\phi^{m+1/2} = \phi^m + \frac{1}{2}\Delta t R^m, \tag{3.6a}$$

$$\phi^{m+1} = \phi^m + \Delta t R^{m+1/2}. \tag{3.6b}$$

The possibilities of utilising implicit, semi-implicit or predictor-corrector methods depends on the nature of each term in the PDE, and especially on its linearity and spatial differential order. For many complex systems, some combination of the aforementioned temporal discretisation schemes will be necessary in order to find a second-order accurate solution which can handle all terms. Furthermore, care must be taken to limit the size of the time step so that the algorithm remains stable and the results accurate. This will be discussed in more detail in sections 4.3 and 5.1.

3.2 Hyperbolic fluxes

The only other terms in hyperbolic PDEs, besides the temporal derivative, can be written either as the divergence of some flux $\boldsymbol{F}$, or as source terms without temporal or spatial derivatives in them. Although we are dealing with PDEs which are not hyperbolic, there are terms in the governing equations which are characteristic of such fluxes. We therefore briefly explain how to deal with these terms. A thorough deliberation on the topic is given in the book by Toro[74].

A source-free hyperbolic system of PDEs for the vector of unknowns $\boldsymbol{q}$ is governed by

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{q}) = 0. \tag{3.7}$$

Conservative update formulas for (3.7) are derived by integrating over a control volume $\Omega_{\boldsymbol{i}}$ and time interval $[t^m, t^{m+1}]$, and applying the divergence theorem to the flux term. Denoting cell-averaged values by $\boldsymbol{q}_{\boldsymbol{i}}(t) = \frac{1}{V_{\boldsymbol{i}}} \int_{\Omega_{\boldsymbol{i}}} \boldsymbol{q}(\boldsymbol{x}, t)\, \mathrm{d}V$, this leads to

$$\boldsymbol{q}_{\boldsymbol{i}}^{m+1} = \boldsymbol{q}_{\boldsymbol{i}}^{m} - \frac{\Delta t}{V_{\boldsymbol{i}}} \int_{\partial \Omega_{\boldsymbol{i}}} \bar{\boldsymbol{F}} \cdot \boldsymbol{n}\, \mathrm{d}S. \tag{3.8}$$

Here, we have introduced the cell volume $V_{\boldsymbol{i}}$ in addition to the time-averaged flux over the time step,

$$\bar{\boldsymbol{F}}(\boldsymbol{x}) = \frac{1}{\Delta t} \int_{t^m}^{t^{m+1}} \boldsymbol{F}(\boldsymbol{x}, t)\, \mathrm{d}t. \tag{3.9}$$

Up to this point, no approximations have been introduced in the scheme; (3.8) is exact. However, this is only true when viewing $\boldsymbol{q}_{\boldsymbol{i}}$ as cell-averaged values; in fact, the evaluation at cell centres is a second-order accurate midpoint integration scheme. The numerical approximations due to discretisation in time and space are all bundled in the evaluation of (3.9). For a discrete number of $N_{\boldsymbol{i}}$ piecewise linear cell faces enclosing the cell, evaluating the flux at the face centres corresponds to second-order midpoint quadratures. Rewriting the spatial integral as a sum over faces, we thus have

$$\boldsymbol{q}_{\boldsymbol{i}}^{m+1} = \boldsymbol{q}_{\boldsymbol{i}}^{m} - \frac{\Delta t}{V_{\boldsymbol{i}}} \sum_{f} A_f \bar{\boldsymbol{F}}_f \cdot \boldsymbol{n}_f, \tag{3.10}$$

and the only remaining challenge is in the evaluation of the time-averaged face-centred fluxes $\bar{\boldsymbol{F}}_f$.

In order to obtain accurate approximations of the different physical variables at cell faces, when their representation in the data is cell-centred, is typically done through the use of slope-based extrapolation and upwinding. This is particularly important in hyperbolic systems, which are characterised by the fact that discontinuities are conserved – this property must be captured in neighbouring cells by the numerical scheme. Here, the term upwinding is used to describe an algorithm which takes the velocity field into consideration when picking data from neighbouring cells. This ensures that the flow of information follows the fluid motion.

Recall that the input data at time t^m are cell averages, i.e. constant within each cell. For each variable $q \in \boldsymbol{q}$, we start by computing the slopes $q_{\boldsymbol{i}}(\boldsymbol{x})$, which are cell-wise linear profiles, although higher order polynomials can also be used, such as in the piecewise parabolic method[75]. Denoting by $\boldsymbol{x_i} = (x_{\boldsymbol{i}}, y_{\boldsymbol{i}}, z_{\boldsymbol{i}})$ the coordinate of the centre of cell $\boldsymbol{i}$, the profile is given by

$$q_{\boldsymbol{i}}(\boldsymbol{x}) = q_{\boldsymbol{i}}^m + \boldsymbol{\delta} q_{\boldsymbol{i}} \odot \frac{\boldsymbol{x} - \boldsymbol{x_i}}{\Delta x}, \tag{3.11}$$

where Δx is the distance between adjacent cell centres. An example of a second-order monotonicity-preserving scheme results from defining the difference approximations

$$\delta_L q_{\boldsymbol{i}} = q_{i+1,j,k}^m - q_{i,j,k}^m, \quad \delta_C q_{\boldsymbol{i}} = \frac{1}{2}(q_{i+1,j,k}^m - q_{i-1,j,k}^m), \quad \delta_R q_{\boldsymbol{i}} = q_{i,j,k}^m - q_{i-1,j,k}^m, \tag{3.12}$$

and limiting the differences by setting

$$\delta_x q_{\boldsymbol{i}} = \operatorname{sgn}(\delta_C q_{\boldsymbol{i}}) \times \begin{cases} \min\left(|\delta_L q_{\boldsymbol{i}}|, |\delta_C q_{\boldsymbol{i}}|, |\delta_R q_{\boldsymbol{i}}|\right), & \text{if} \quad \delta_L q_{\boldsymbol{i}} \delta_R q_{\boldsymbol{i}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Similar difference approximations must naturally be computed in the other directions too. We then have an overall distribution which is globally piecewise linear, as exemplified in a single dimension in figure 3.1.

Since we have the slopes $q_{\boldsymbol{i}}$ within each cell, we can extrapolate q to each cell face from both of its neighbouring cells. Note that on a normal cell face, two of the three centroid coordinates are equal to that of its neighbouring cell centres. For example, consider the interface between cells (i, j, k) and $(i+1, j, k)$. The face centroid is located at $(x_i + \frac{1}{2}\Delta x, y_j, z_k)$, so we refer to it by $i + \frac{1}{2}$. Two extrapolated approximations of q at the face, based on the cells to its left and right, are

$$q_{i+\frac{1}{2},L} = q_{i,j,k}^m + \frac{1}{2}\delta_x q_{i,j,k} \quad \text{and} \quad q_{i+\frac{1}{2},R} = q_{i+1,j,k}^m - \frac{1}{2}\delta_x q_{i+1,j,k}, \tag{3.14}$$

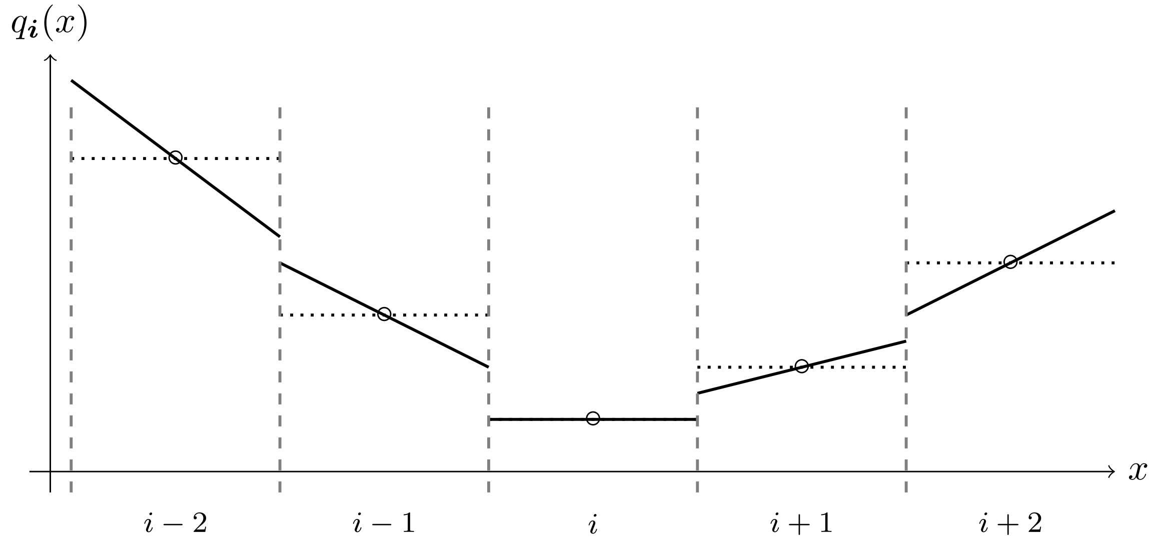


Figure 3.1: Illustration of piecewise linear slopes, reconstructed from spatial averages within each computational cell. For clarity, this figure shows the simplified case of a single spatial dimension.

respectively. In order to choose which of these approximations is best, a simple upwinding algorithm like that shown in algorithm 3.1 is used. Given extrapolated values from the two different sides of the interface, and an advection velocity u_{adv} in the direction normal to the interface, an appropriate choice can be made. If the advection velocity is positive, the left state is chosen, while the right state is chosen if the opposite is true. In the event that $|u_{\text{adv}}| = 0$ (i.e. smaller than some numerical tolerance), the average of the two states is taken.

Algorithm 3.1 Simple upwinding

```
function UPWIND($q_L, q_R, u_{\text{adv}}$)
    if $u_{\text{adv}} = 0$ then
        return $\frac{1}{2}(q_L + q_R)$
    else if $u_{\text{adv}} > 0$
        return $q_R$
    else
        return $q_L$
    end
end function
```

There is one missing ingredient preventing us from applying the upwinding scheme to find face-based values for all variables, namely the spatially staggered advection velocity $\boldsymbol{u}_{\text{adv}}$. By spatially staggered, we mean that each of its components is stored on the face with normal vector parallel to it. We obtain this field from the extrapolated normal components of the velocity field at each face, using algorithm 3.1 with the advection velocity taken as the average of the left and right states:

$$u_{\mathrm{adv,i}+\frac{1}{2}} = \textsc{Upwind}\left(u_{i+\frac{1}{2},L}, u_{i+\frac{1}{2},R}, \frac{1}{2}\left(u_{i+\frac{1}{2},L} + u_{i+\frac{1}{2},R}\right)\right). \tag{3.15}$$

With the advection velocity available, all other extrapolated variables can be upwinded at the cell faces, and hyperbolic fluxes can be evaluated with high precision while preserving monotonicity. As mentioned in the beginning of this section, there is a large amount of literature describing how this simplest of algorithms can be expanded upon, which has been studied so extensively due to the importance in purely hyperbolic systems of PDEs.

3.3 Projection methods for incompressible flow

A major difficulty in numerical solutions of incompressible flow problems it the fact that the pressure and velocity fields are coupled to each other through the incompressibility constraint $\nabla \cdot \boldsymbol{u} = 0$, rather than through a classical energy conservation equation. This coupling makes it impossible to solve the equations directly, even numerically. A common way to solve the transient equations is to use projection methods. Projection methods decouple the interdependencies of pressure and velocity to a sequence of elliptic problems within each time step. Consequently, the algorithms can yield efficient simulations for large systems[76].

The original projection method for the incompressible Navier-Stokes equations was developed by Chorin in 1968[77], and the essence is still the same in most modern variants. Chorin's projection method is an operator splitting approach, in which, for each time step, one first performs an intermediate time step for the velocity field while ignoring the pressure forces, and then enforces incompressibility in order to update the pressure. Finally, the velocity can be updated. Because of the intermediate step in the algorithm, it is commonly referred to as a fractional step method. However, it does not fit into the original class of methods called fractional step or splitting schemes[78]. This is due to the fact that, unlike in compressible systems, the pressure is not a dynamic variable, but takes the role of a Lagrange multiplier. The resulting PDE is therefore not of Cauchy-Kowalevski type, and corresponding algorithms are not generalisable to higher orders of accuracy[79]. This is one of the main reasons why there has been such a large focus on the development and analysis of accurate, efficient and scalable projection methods in the last fifty years.

The family of numerical algorithms known as projection methods can generally be divided into three groups. In the first group, exemplified by the pioneering work of Chorin, the pressure gradient is ignored in the first step of the algorithm, and subsequently corrected for afterwards. They are therefore called pressure-correction methods. Velocity-correction methods, on the other hand, start by ignoring the viscous term, and instead correct for this afterwards. More recently, so-called Gauge methods[80;81], and more generally consistent splitting methods[82] have been introduced, which are free of the splitting errors associated with the former two.

The Helmholtz decomposition[83] allows us to write any vector field as a sum of solenoidal and irrotational parts, i.e.

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} - \nabla\Phi, \tag{3.16}$$

where $\boldsymbol{u}$ has zero divergence and $\nabla\Phi$, being a scalar gradient, is irrotational. This is the building block of projection methods, where the goal is to find the divergence-free velocity field $\boldsymbol{u}$ associated with a generic approximation $\tilde{\boldsymbol{u}}$, which has an irrotational part. Taking the divergence of (3.16), leads to a Poisson equation for Φ,

$$\nabla^2\Phi = -\nabla\cdot\tilde{\boldsymbol{u}}. \tag{3.17}$$

which is straightforward to solve since the right-hand side is known. In turn, the solution allows us to compute the divergence-free velocity field $\boldsymbol{u}$, according to (3.16). This simple example shows how we can project any solution onto the space of divergence-free vector fields.

Although the above illustrates the basic tricks utilised in projection methods, it doesn't explicitly show how the velocity and pressure fields in Cauchy's momentum balance are decoupled from each other. This is the operator splitting part of the algorithm, and it means that we can update the pressure at the same time as the velocity. As an example, consider the MOL discretised version of (2.29b):

$$\frac{\boldsymbol{u}^{\text{new}} - \boldsymbol{u}^{\text{old}}}{\Delta t} = -\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \frac{1}{\rho}\left(-\nabla p + \nabla\cdot\boldsymbol{\tau} + \boldsymbol{f}\right), \tag{3.18}$$

Following Chorin's original approach, we can introduce an intermediate velocity field $\tilde{\boldsymbol{u}}$. Then the two equations

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^{\text{old}}}{\Delta t} &= -\boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \left(\nabla \cdot \boldsymbol{\tau} + \boldsymbol{f} \right) && \text{(3.19a)} \\ \frac{\boldsymbol{u}^{\text{new}} - \tilde{\boldsymbol{u}}}{\Delta t} &= -\frac{1}{\rho} \nabla p && \text{(3.19b)}\end{aligned}$$

sum to (3.18). The former can be solved for $\tilde{\boldsymbol{u}}$ without worrying about the coupling to pressure, and the latter is of the form of a Helmholtz decomposition, just like (3.16). Taking its divergence, we can therefore solve the Poisson equation

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{\text{new}} \right) = \nabla \cdot \tilde{\boldsymbol{u}}, \tag{3.20}$$

for p^{new}, and obtain the new, divergence-free velocity field

$$\boldsymbol{u}^{\text{new}} = \tilde{\boldsymbol{u}} - \frac{\Delta t}{\rho} \nabla p^{\text{new}}. \tag{3.21}$$

As we will show, some higher-order schemes require a projection to be applied at an intermediate time step in order to ensure a divergence-free partially evolved velocity field. This is in addition to the projection for the final velocity. Projection methods constitute a rich field of research, and there are many subtleties connected to their implementations and convergence and stability analysis. For an excellent overview, we recommend the review paper by Guermond et al.[76]

Projection methods for steady-state flows

A popular family of algorithms for solving nonlinear systems of the form that arises from our spatial discretisation of (2.30) at steady-state, is the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)[84]. Rather than split each time step into two decoupled equations, the basic idea in these methods is to construct a number of linear systems to be solved iteratively until convergence is achieved[85]. Linearization of the discretised sets of equations is done by computing some of the terms with values from previous iterations. For a full explanation of the method, we recommend Zikanov's book[86], which also covers extensions of the method such as SIMPLER[87] and SIMPLEC[88].

3.4 Non-Newtonian fluids

Just as for Newtonian fluids, the pursuit of knowledge about viscoplastic fluids has relied heavily on computational methods in the last fifty years. Compared to the Newtonian case, however, the numerical simulations are much more demanding in terms of processing time. This is largely due to the existence of a yield stress, since this results in a singularity at zero strain rate for the apparent viscosity of the fluid. Traditional methods for computing the solution are rendered useless for this case, since infinite viscosities cannot be represented in the unyielded regions. There are now two main branches of algorithms designed to deal with this problem. The first utilises mathematical regularisation to approximate the viscosity function in the low-strain limit. In doing so, the rigid body approximation is effectively replaced by a fluid with very high viscosity in the unyielded regions. A regularisation parameter controls how close this approximation is to the idealised viscoplastic case. On the other hand, the problem can be reformulated in the framework of non-smooth optimisation theory, and solved using augmented Lagrangian methods. Such methods can solve the unregularised problem without introducing approximations, but require significantly more computational resources to find the solution[89].

The regularisation approach was first explored by Bercovier and Engelman in 1980[90], who utilised a simple yet efficient work-around by adding a small constant to the computed strain rate, so that even in the zero-strain limit the viscosity would remain finite. An alternative method proposed by Tanner and Milthorpe a few years later was the bi-viscosity model[91], in which the viscoplastic fluid is characterised by a separate, large viscosity when the strain rate is below a given threshold. An exponential regularisation factor was then introduced by Papanastasiou in 1987[92], and this method of regularisation is still widely used in modern codes which deal with viscoplasticity through regularisation. Important investigations based on Papanastasiou regularisation include those of Mitsoulis et al.[93–96] and Syrakos et al.[64;97;98]

Augmented Lagrangian methods were first applied to the viscoplastic flow problem in 1983 by Fortin and Glowinski[99], but the variational formulation on which it relies was derived by Duvaut and Lions in 1976, who studied existence, uniqueness and regularity of solutions to the problem[100]. The augmented Lagrangian algorithm itself is due to Hestenes (1969)[101]. Although the simulation of viscoplastic flow using this optimisation technique constitutes an important milestone in the history of its numerical treatments, the regularisation approach was much more popular due to the large discrepancy in computational resource requirements. Advances in convex optimisation over the next few decades led to the work of Saramito and Roquet[102], where significant improvements were achieved in terms of convergence rates, and hence, computational efficiency. Other contributions[103] further confirmed the potential, and in recent years state-of-the-art algorithms are actively being developed, notably by Treskatis et al.[104], Saramito[105], Bleyer[106] and Dimakopoulos et al.[107] Due to the ability to solve the unregularised viscoplasticity problem, these methods have become increasingly popular since algorithmic advances have led to significant speed-up of their runtime. It is important to note, however, that these methods are still much more costly than regularisation approaches, so although they allow computation of exact locations of yield surfaces, they are unnecessarily expensive when this is less important, and a general understanding of the flow field is desirable. An example is within cement displacement complexities for engineering purposes[7], where the desirable insight is pressure distributions and the cement displacement efficiency[16].

Two excellent review papers on advances in numerical simulations of these fluids appeared in a recent special edition of Rheologica Acta[89;96].

Regularisation of apparent viscosity

Equation (2.27) does not present any problems for unyielded regions ($\|\dot{\boldsymbol{\gamma}}\| > 0$), but the apparent viscosity has a singularity when the strain rate approaches zero (and $\|\boldsymbol{\tau}\| \to \tau_0$). Computational schemes such as those implemented in IAMR cannot be used in the presence of such singularities. Regularisation deals with the problem by replacing the ill-behaved apparent viscosity with a function that approximates the rheological behaviour, but which stays bounded for arbitrarily small $\|\dot{\boldsymbol{\gamma}}\|$. This is done by introducing an additional parameter ε to the apparent viscosity, which describes how big the effect of the regularisation is. A large value of ε allows for inexpensive computations even near unyielded flow, while the limit $\varepsilon \to 0$ recovers the unregularised description. Note that one must be careful to choose a small enough value for ε, in addition to fine enough mesh resolutions and convergence criterion. Otherwise, the mathematical approximation will not hold.

Consider the Bingham model as given in (2.25). The simplest regularisation possible would be to just let $\|\dot{\boldsymbol{\gamma}}\| \to \|\dot{\boldsymbol{\gamma}}\| + \varepsilon$, leading to an apparent viscosity function

$$\eta = \mu + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\| + \varepsilon}. \tag{3.22}$$

This regularised apparent viscosity clearly removes the singularity in the manner outlined above, and recovers the Bingham model in the limit $\varepsilon \to 0$. An improved regularisation was proposed by Bercovier and Engelman[90], in which one instead lets $\|\dot{\gamma}\| \to \sqrt{\|\dot{\gamma}\|^2 + \varepsilon^2}$. With this regularisation, the apparent viscosity becomes

$$\eta = \mu + \frac{\tau_0}{\sqrt{\|\dot{\gamma}\|^2 + \varepsilon^2}}. \tag{3.23}$$

Finally, a very popular regularisation was introduced by Papanastasiou[92] in 1987. This method, which we will be utilising, introduces an exponential relaxation according to

$$\eta = \mu + \frac{\tau_0}{\|\dot{\gamma}\|}\left(1 - e^{-\|\dot{\gamma}\|/\varepsilon}\right). \tag{3.24}$$

This is a good approximation for $\|\dot{\gamma}\|/\varepsilon \gg 1$, while in the small strain limit we have

$$\lim_{\|\dot{\gamma}\| \to 0} \frac{1 - e^{-\|\dot{\gamma}\|/\varepsilon}}{\|\dot{\gamma}\|} = \lim_{\|\dot{\gamma}\| \to 0} \frac{1}{\|\dot{\gamma}\|}\left(1 - \sum_{k=0}^{\infty} \frac{(-\|\dot{\gamma}\|/\varepsilon)^k}{k!}\right) = \frac{1}{\varepsilon}, \tag{3.25}$$

so that it always remains bounded, and recovers the unregularised model in the limit $\varepsilon \to 0$. The effect of varying the regularisation parameter ε is shown in Figure 3.2. In the Papanastasiou-regularised Bingham model, the singularity in apparent viscosity is replaced by the limiting value

$$\lim_{\|\dot{\gamma}\| \to 0} \eta = \mu + \frac{\tau_0}{\varepsilon}. \tag{3.26}$$

Since the regularisation parameter has units s^{-1} like the strain rate, we introduce the dimensionless Papanastasiou number

$$Pa = \frac{\mathcal{U}}{\varepsilon \mathcal{L}}, \tag{3.27}$$

which measures the degree of regularisation employed in the apparent viscosity, with higher Pa being a more accurate description of unregularised viscoplasticity. In dimensionless form, (3.24) is then

$$\hat{\eta} = 1 + \frac{Bi}{\|\hat{\dot{\gamma}}\|}\left(1 - e^{-Pa\|\dot{\gamma}\|}\right), \tag{3.28}$$

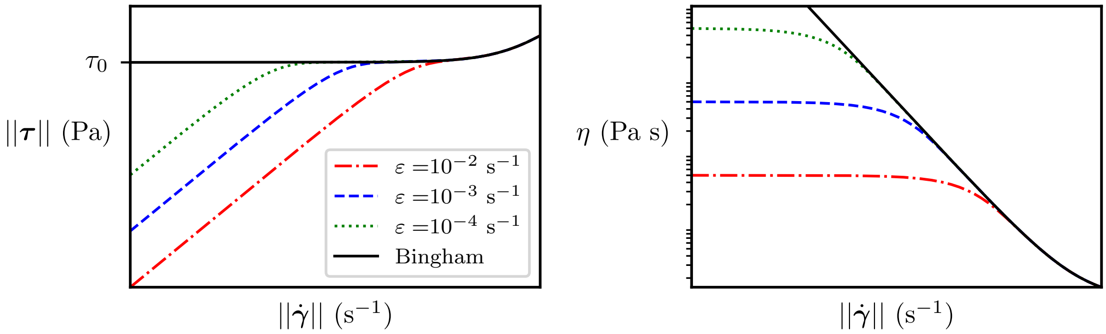


Figure 3.2: Papanastasiou regularisation: stress magnitude (left) and apparent viscosity (right) as functions of the magnitude of the rate-of-strain tensor $\dot{\boldsymbol{\gamma}}$. Note that both plots utilised logarithmic axes. For decreasing ε, we recover a closer approximation to the actual Bingham model.

while the low-strain limiting value is

$$\lim_{\|\hat{\dot{\boldsymbol{\gamma}}}\| \to 0} \hat{\eta} = 1 + Bi\, Pa. \tag{3.29}$$

Combination of Papanastasiou regularisation with the Herschel-Bulkley model as given in (2.27), gives the regularised apparent viscosity as

$$\eta = \left(\kappa \, \|\dot{\boldsymbol{\gamma}}\|^{n-1} + \frac{\tau_0}{\|\dot{\boldsymbol{\gamma}}\|} \right) \left(1 - e^{-\|\dot{\boldsymbol{\gamma}}\|/\varepsilon} \right). \tag{3.30}$$

In dimensionless form, (3.30) becomes

$$\hat{\eta} = \left(\left\| \hat{\dot{\boldsymbol{\gamma}}} \right\|^{n} + HB \right) \frac{1 - e^{-Pa \|\hat{\dot{\boldsymbol{\gamma}}}\|}}{\left\| \hat{\dot{\boldsymbol{\gamma}}} \right\|}. \tag{3.31}$$

A note on regularisation methods

With regularisation in place, it is straightforward to compute numerical solutions to (2.29) in conjunction with the viscoplastic constitutive equation (3.30) by directly discretising the equations in space and time before utilising standard methods from a toolbox for PDEs. It is important to note, however, that we are actually solving a slightly different problem as long as $\varepsilon > 0$. The last couple of decades has seen a wide discussion on the benefits and pitfalls of approximating the single-parameter Bingham model through regularisation. On the one hand, there is a fundamental question of $\pi\alpha\nu\tau\alpha\,\rho\epsilon\iota$ – whether "*everything flows*". In their provocative paper from 1985[62;108–110], Barnes and Walters claimed that yield stress fluids did not actually have a non-zero yield stress, but that this was an extrapolated value due to the very rapid increase in viscosity at small strain rates for such fluids. In this context, modern models such as that by de Souza Mendes and Dutra[60] actually strive to capture the rapid rate of change through an exponential just like the Papanastasiou regularisation. Advocates of the pure Bingham model are of the opinion that a single-parameter model is more elegant and better suited mathematically to capture the essence of yield stress influence on fluid behaviour. With this in mind, there is a second question of which numerical techniques are appropriate to utilise in order to obtain an accurate understanding of the fluids in question. Papanastasiou regularisation was not intended to capture small-strain effects of yield stress fluids better than the Bingham model, but rather intended to circumvent a numerical challenge[111]. Although it does this well, contemporary algorithms from the field of non-convex optimisation are able to solve the pure Bingham problem, albeit at a high computational expense. In spite of the complexity and runtime costs of such algorithms, there has been a steadily growing group of academics who support this as the best way forward for simulating viscoplastic fluid flow. We will presently give a brief introduction to the relevant mathematical formulation and algorithms, but in any case, Papanastasiou regularisation is extensively used in state of the art fluid mechanics codes and contemporary research, and the only viable option for our purposes.

Variational formulation and convex optimisation

For the unregularised problem, the relevant algorithms are instead based on optimisation methodology. This is due to the realisation that there is a strong resemblance between the viscoplastic flow problem and convex optimisation as utilised in image analysis. Consequently, several connections between these two fields have been made, leading to sophisticated algorithms with impressive results. In order to see why such methods can be applied to the problem at hand, we briefly explain the derivation of a variational form, and which optimisation problem characterises its solution.

Consider the strong form of the problem at hand: Find $(\boldsymbol{u}, p)$ subject to (2.29b), (2.29e) and (2.27) in Ω and $\boldsymbol{u} = \boldsymbol{u}_{BC}$ on $\partial\Omega$. Additionally, the problem is defined by a suitable initial condition. We wish to find a weak form of the problem by majorising the virtual viscous power dissipated in the domain. Introducing a test function $\boldsymbol{v}$ in an appropriate function space and integrating over Ω, we then obtain the variational inequality[100]

$$\rho \int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) \cdot (\boldsymbol{v} - \boldsymbol{u}) \,\mathrm{d}V + \kappa \int_\Omega \|\mathcal{D}\boldsymbol{u}\|^{n-1} \mathcal{D}\boldsymbol{u} : (\mathcal{D}\boldsymbol{v} - \mathcal{D}\boldsymbol{u}) \,\mathrm{d}V \\ + \tau_0 \int_\Omega (\|\mathcal{D}\boldsymbol{v}\| - \|\mathcal{D}\boldsymbol{u}\|) \,\mathrm{d}V \geq \int_\Omega \boldsymbol{f} \cdot (\boldsymbol{v} - \boldsymbol{u}) \,\mathrm{d}V, \quad (3.32)$$

where we are using the notation $\mathcal{D}\boldsymbol{v} = \left(\nabla + \nabla^\top\right) \boldsymbol{v}$, so that $\dot{\gamma} = \mathcal{D}\boldsymbol{u}$. We refer to $\boldsymbol{u}(\boldsymbol{x}, t)$ as a weak solution of the unsteady viscoplastic flow problem if it satisfies (3.32). The most prominent algorithms in the literature divide the problem into a stationary problem and a temporal marching scheme. At steady-state, the momentum balance is given by (2.30), so that the variational inequality reduces to

$$\kappa \int_\Omega \|\mathcal{D}\boldsymbol{u}\|^{n-1} \mathcal{D}\boldsymbol{u} : (\mathcal{D}\boldsymbol{v} - \mathcal{D}\boldsymbol{u}) \,\mathrm{d}V + \tau_0 \int_\Omega (\|\mathcal{D}\boldsymbol{v}\| - \|\mathcal{D}\boldsymbol{u}\|) \,\mathrm{d}V \geq \int_\Omega \boldsymbol{f} \cdot (\boldsymbol{v} - \boldsymbol{u}) \,\mathrm{d}V. \quad (3.33)$$

Timm Treskatis gives an excellent introduction to the connection between convex optimisation and viscoplastic flow in his Ph.D. thesis[112]. In it, he introduces a common problem in the field of image analysis, namely deblurring of imperfect images. Denoting the undisturbed image by $\boldsymbol{x}$, the optimisation approach seeks to minimise the sum of a convex functional F and a regularisation term R weighted by a factor α:

$$\boldsymbol{x} = \arg\min \left(F(\boldsymbol{x}) + \alpha R(\boldsymbol{x}) \right). \quad (3.34)$$

This regularisation term will often be a function of the magnitude of the gradient of $\boldsymbol{x}$, in order to penalise noise and blur in images while allowing large areas of constant colour. Now the steady-state variational inequality as given in (3.33) has the characteristic property that a velocity field $\bar{\boldsymbol{u}}$ is a weak solution to it if

$$\bar{\boldsymbol{u}} = \arg\min \left(\kappa \int_\Omega \|\mathcal{D}\boldsymbol{u}\|^{n+1} \,\mathrm{d}V - \int_\Omega \boldsymbol{f} \cdot \boldsymbol{u} \,\mathrm{d}V + \tau_0 \int_\Omega \|\mathcal{D}\boldsymbol{u}\| \,\mathrm{d}V \right). \quad (3.35)$$

Such a solution exists and is unique. One can immediately see the connection between (3.35) and the example problem from convex optimisation. The yield stress τ_0 thus serves as a weighting factor for the regularisation term in the minimisation algorithm. Due to this insight, many advanced algorithms from convex optimisation are made available for the solution of the viscoplastic flow problem.

Chapter 4

High-performance computing

In essence, CFD is the science of calculating quantitative predictions of physical systems involving fluid flow, by using modern computers[113]. CFD software generally falls into one of three main categories. The first of these is generic commercial software, which has versatile coverage for the most common flows of interest, and often also for other continuum problems. However, these code often focus on Computer Aided Design (CAD) and simple Graphical User Interfaces (GUIs), and hide the numerical schemes and algorithmic detail under the hood. Examples include the ANSYS suite[114], COMSOL Multiphysics[115] and Siemens Star-CCM+. The second class of CFD codes is commercial service-provider software, which has been tailor-made for a specific problem, and which is inflexible in terms of modification and augmentation. Finally, we have open-source and academic software, which is generally state-of-the-art in a few specific metrics, but otherwise demanding to use for commercial operation purposes due to the focus on mathematical robustness and continuous implementation of the latest published technology, with less regard to ease of use for non-experts. Examples of these include OpenFOAM[116], FEniCS[117;118] and the suite of codes based on the AMReX framework[17].

It would not be wise to put all of the above in the same class and make generic statements about their validity, because it all depends on what systems of equations they are solving, the numerical methods and techniques they are employing and their effectiveness of using current computational hardware architectures. For the type of development necessary in this project, it is only the latter class of CFD codes which is really relevant. For example, we require the flexibility of open-source software, so that we can access and alter any submodule in the code, in addition to writing our own modules. At the same time, we are in a much better position to discuss the problems in the wider scientific community when opting for this kind of tool. Most importantly, however, we need to ensure that the software suite which we choose to base our development upon is compatible with the latest algorithmic and hardware developments in high performance computing, since scalability is of the utmost essence for the project.

4.1 Parallel processing and scalability

Among the most important attributes of High Performance Computing (HPC) codes, is their intrinsic scalability on state-of-the-art computer architectures. Due to major advances within parallel computing in the last few decades, it has been important to discuss which methods of evaluating their capabilities provide the best insight. Moreland and Oldfield provide an excellent overview of contemporary formal metrics for such measurements[119].

Suppose we are trying to solve a computational problem of size N. Let $T^*(N)$ denote the execution time of the best serial algorithm, while $T(N, P)$ is the time used by a parallel algorithm deployed on P processor cores. Then the speed-up of the parallel algorithm is given by

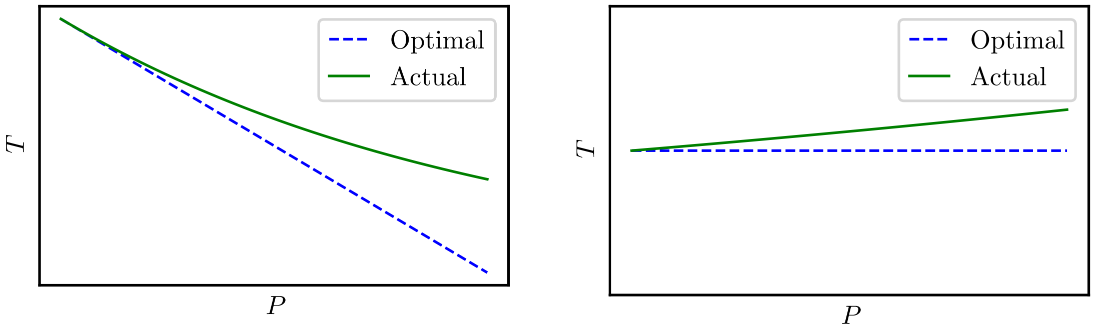
$$S(N, P) = \frac{T^*(N)}{T(N, P)}. \tag{4.1}$$

Note that in practice, it is common to use $T^*(N) = T(N, 1)$, although this is not entirely appropriate if there exists faster serial algorithms. The theoretically best possible speed-up occurs if the entire task can be divided into as many chunks of equal work as there are cores being used, so that it is distributed among the available cores without any overhead. In this utopian case, we have speed-up equal to the processor core count, $S(N, P) = P$. As a measure of the efficiency $E(N, P)$ of the parallel algorithm, we can therefore take the ratio of the observed speed-up to the ideal, i.e.

$$E(N, P) = \frac{S(N, P)}{P} = \frac{T^*(N)}{PT(N, P)}. \tag{4.2}$$

Although both speed-up and efficiency are useful metrics for analysing the parallel performance of an algorithm, they have one caveat. Since the serial runtime is necessary in order to compute them, it quickly becomes impractical for analysis of large-scale applications. Computing the serial runtime for very large N takes far too long. For the metrics presented thus far, the problem size is kept fixed, and the amount of allocated resources increased. Such scaling is known as strong scaling.

On the other hand, the problem size can be scaled so that it is proportional to the core amount P. If one core is used to solve the problem with size N^*, one would then ideally hope that the parallel runtime with P processors for solving a problem P times as large, $T(PN^*, P)$ remains constant for increasing P. This type of analysis is called weak scaling. Figure 4.1 illustrates the optimal runtime for the two cases, in addition to what typical scaling results look like.



(a) Strong scaling, N constant

(b) Weak scaling, $N \propto P$

Figure 4.1: Optimal and typical runtime results of parallel algorithms measured through strong (left) and weak (right) scaling analysis.

Finally, it may in some case be difficult to keep the problem size proportional to the job size, so that weak scaling analysis becomes difficult to obtain. However, one can measure the amount of work done per time unit, referred to as the rate:

$$R(N, P) = \frac{N}{T(N, P)}. \tag{4.3}$$

The rate removes the weak scaling dependency between problem size and core amount. Sampling the rate across a selection of problem sizes and core amounts can thus help the user pick runtime parameters which yield high computational efficiency.

4.2 The AMReX framework

Based on our requirements for structured AMR, parallelisable linear solvers and scalability, the most complete, advanced and efficient software framework is AMReX[17], which is an open-source project funded by the United States Department of Energy

(DOE) through the Exascale Computing Project. Exascale computing refers to supercomputing capabilities exceeding 10^{18} floating point operations per second (exaFLOPS), and it is the current limit of modern supercomputers[120]. The fact that AMReX is the DOE's back-end framework of choice for solving multiphysics PDEs on emerging architectures, combined with its extensive list of tools and features, made it an easy choice for our starting point.

AMReX is a mature software framework for building massively parallel block-structured AMR applications for the numerical solutions of PDEs[17]. It is developed at the Center for Computational Sciences and Engineering (CCSE) at Lawrence Berkeley National Laboratories (LBNL), and is written in hybridised C++ and Fortran 90. AMReX facilitates a variety of approaches to parallelisation, including straightforward utilisation of e.g. MPI, OpenMP and, increasingly, the use of GPUs through the CUDA language. More generally, AMReX facilitates a hybrid parallelisation model, where individual grids are distributed to MPI ranks, and a secondary tool (MPI / OpenMP / CUDA etc.) is used to thread over logical tiles within the grids. This allows users a lot of flexibility to tailor the parallelisation approach to their specific use case. Importantly, all functionality is written such that the software can be deployed on machines ranging from old laptops to emerging supercomputing clusters. Applications based on AMReX have demonstrated excellent strong and weak scaling up to hundreds of thousands of cores[121–124].

The framework contains extensive software support and tools for implementing algorithms which solve PDEs with complex boundary conditions, all built to work with a range of discretisation methodologies for block-structured AMR, both in space and for temporal evolution. It is possible to design customised approaches for each refinement level, or to utilise the automatic subcycling and multi-level synchronisation which can be directly applied to the full hierarchy of grids. All user-defined code can be written in either C++ or Fortran (or a combination thereof), and can utilise built-in iterators and data types to obtain information about, and operate on, specific patches of the discretised domain. These data types and iterators are specifically built to act as a solid foundation for the AMR and parallelisation approaches. Variables stored in the structured grid can be defined at a range of different locations within the cells, depending on the dimensionality of the problem. For fully 3D applications, the choices are at cell centres, cell faces, cell edges or cell corners (nodes). Additionally, it is possible to utilise one-dimensional spherical coordinates or two-dimensional polar coordinates in addition to standard Cartesian ones.

Among the more advanced features which are of utmost importance for PDE solvers with elliptic or parabolic subproblems, are the linear solvers included in AMReX. A range of different preconditioners and iterative solution methods are implemented, including multilevel geometric multigrid solvers[125;126] with the biconjugate gradient stabilised method[127] or other Krylov solvers available for use at the coarsest multigrid level. There are also built-in tools which allow the representation of complex boundaries through embedded boundaries and cut cells. Although the user is required to write specific submodules for how to treat cut cells, all necessary data is readily available through the data structure interfaces, and many complex operations, such as linear solvers, have separate implementations which automatically take care of the necessary changes.

From a practical point of view, the code includes native input and output routines, both for writing results to file and for checkpoint/restart capabilities, allowing simulations to be paused and re-started at the convenience of the user. AMReX' native output format is supported by well-known visualisation software such as Paraview[128], yt[129] and VisIt[130], and can also be converted to HDF5. There is also a custom performance profiling tool specialised for AMReX applications, which takes into account the data types and iterators used for block-structured AMR data.

In addition to the above, there are several features of AMReX which we will not be taking advantage of in our work. This includes separate data structures for particles, which can be used for particle-in-cell methods, as Lagrangian tracer particles etc. The code features interoperability with other software for scientific computing such as PETSc[131] and hypre[132], which is another reason for its popularity. Among open-source codes which are based on AMReX, there are examples from the fields of accelerator modeling (WarpX)[133], compressible astrophysics (CASTRO)[121], low-Mach-number astrophysics (MAESTRO)[134], cosmology (Nyx)[122] compressible and low-Mach-number combustion (the Pele suite)[135;136], and multiphase flow (MFIX-Exa)[137].

Structured adaptive mesh refinement

For a vast range of flow problems, large computational domains are required even though we are only interested in small areas of the domain such as boundaries, interfaces or areas where variables exhibit complex dynamic behaviour. Refining the computational mesh in these regions is a well-known technique in order to save memory and processing power, and it can be applied with various degrees of sophistication. Within the AMReX framework, hierarchical block-structured AMR is employed, which allows for efficient local refinement by symmetrically splitting each cell into 2^d smaller ones, where d is the dimensionality of the problem. Furthermore, the dynamic refinement is spatio-temporal, meaning that we employ subcycling in the discrete time steps to avoid setting global time step restrictions based on the smallest cells. This is crucial for dynamic AMR to be used efficiently. Tagging criteria are utilised in order for the code to determine which cells should be split or merged at a given refinement step, and these can depend on the values of any directly simulated or derived variables, in addition to spatial position and time. We will presently go through the fundamentals of the AMR and subcycling routine, and refer to the original paper by Almgren et al.[138] for further detail.

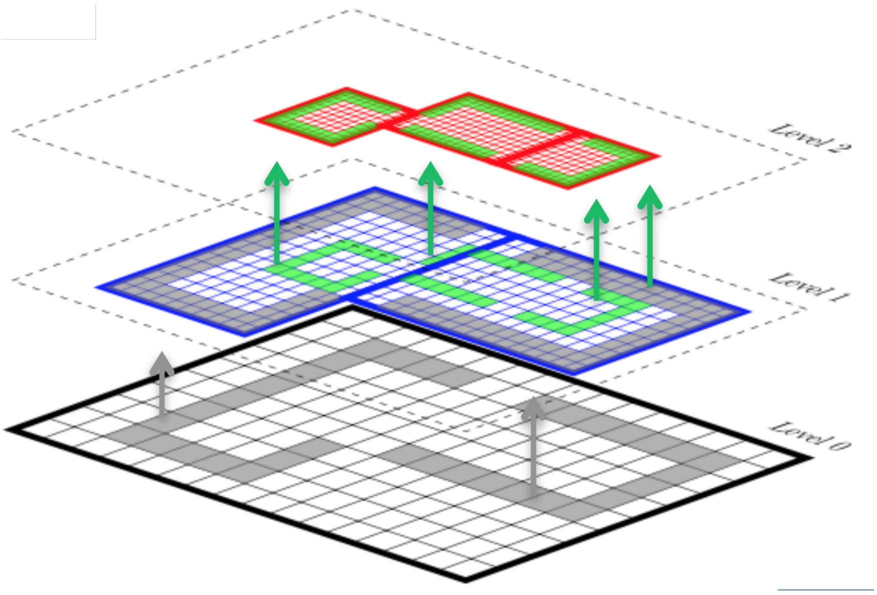


Figure 4.2: Multiple refinement levels, with finer patches utilising coarse level data at boundaries. Image courtesy of Andrew Myers, AMReX developer.

In hierarchical block-structured AMR, the different refinement levels make up a nested grid of rectangular patches. Each level is made up of the union of patches with the corresponding refinement, which do not need to be directly connected to each other. We refer to the levels by their identifier ℓ, and set $\ell = 0$ for the coarsest grid, which must cover the entire domain. There are $\ell_{\max}$ levels in total, so that $0 \leq \ell < \ell_{\max}$, but we note that $\ell_{\max}$ can change dynamically during the simulation, although restricted by a user-specified allowable maximum. As described earlier, refinement is obtained through even partition of the relevant cells, i.e. $\Delta x_{\ell+1} = \frac{1}{r}\Delta x_\ell$, where r is the refinement ratio. Every few steps (the regridding frequency is tunable), the solution is analysed in order to see if the grids need to be updated. Refinement is based on some user-specified criterion, which can be anything from a threshold value on a variable (primary or derived), to tolerance on an error estimate or even a simple geometrical condition. If a cell on level $\ell_{\max}$ is tagged for refinement, then a new refinement level is created, as long as the maximum depth hasn't been reached. Tagged cells at each level of refinement are clustered into rectangular patches which satisfy criteria relating to *max size*, *blocking factor* and *grid efficiency*. Here, the max size is the largest amount of cells along each side of the rectangle, and that length must be divisible by the blocking factor. The grid efficiency describes the percentage of cells within a patch which have been tagged according to the refinement criterion, and is typically expected to exceed 70%. This is done using the Berger-Rigoutsis algorithm[139], which finds non-overlapping sets including all tagged cells which satisfy the above criteria, and chooses the one which minimises the number of new grids that must be created, since the computational overhead between refinement levels increases with the latter. One result of this patch clustering algorithm with relaxed grid efficiency is that a single refinement criterion can be used across levels, while still causing a smaller and smaller part of the domain to be refined for each level. The levels are properly nested, meaning that all grids on level $\ell + 1$ are contained in level ℓ grids. Additionally, there is at least one layer of coarse level cells surrounding the finer patch, so that the coarse level data can be used as Dirichlet boundary conditions for the fine level patches. Figure 4.2 illustrates the refinement hierarchy.

Since the size of the spatial cells directly affects the largest allowable stable time step, the spatial refinement necessitates time step reduction as well. In order to avoid reducing the time step on all levels to that of the finest grid, we employ a subcycling algorithm which ensures that the time step is small enough at each level, while simultaneously keeping the data synchronised within each level. At level $\ell = 0$, the system is incremented by Δt as computed by a CFL-type condition. For $\ell > 0$, it is necessary to take 2^ℓ time steps, and synchronisation must be performed with level $\ell + 1$ every single step and with level $\ell - 1$ every two steps. By synchronisation, we mean extrapolation from coarse-cell data to fine-cell boundaries, and averaging of underlying fine-cell data to coarser cells. The recursive subcycling algorithm is exemplified in figure 4.3.

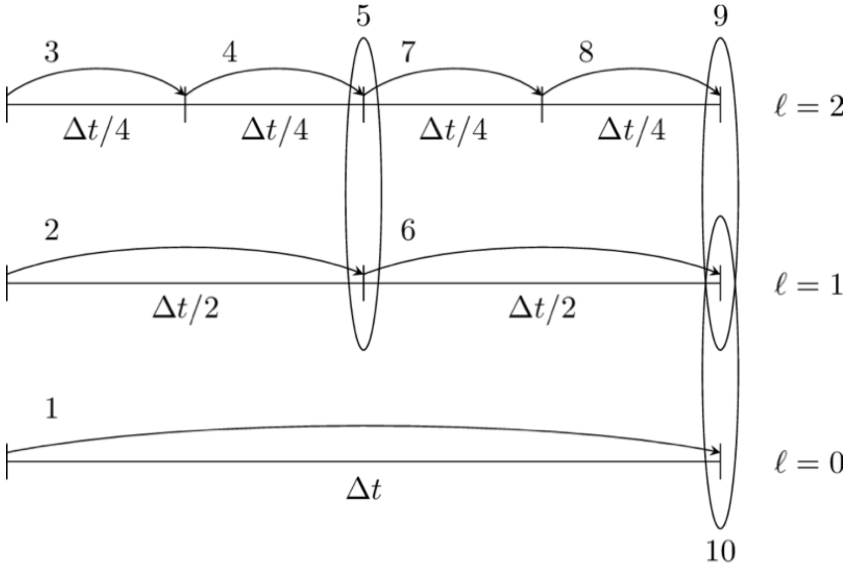


Figure 4.3: Illustration of the temporal subcycling procedure. 1: a full time step Δt is taken on the coarsest grid ($\ell = 0$). 2: a half time step is taken on level $\ell = 1$ and 3-4: two quarter time steps on level $\ell = 2$, before 5: levels 1 and 2 are synchronised. 6-9: Steps 2-5 are repeated to reach the final time on $\ell = 1$ and $\ell = 2$. 10: the coarsest level ($\ell = 0$) is synchronised with the results on the finer grids. Image courtesy of Andrew Myers, AMReX developer.

4.3 IAMR

An approximate projection method for solving the variable-density incompressible Navier-Stokes equations on an adaptive mesh hierarchy has been implemented in the IAMR (Incompressible Adaptive Mesh Refinement) code. The algorithm was described for the constant viscosity case by Almgren et al. [138], with extensions to low-Mach-number reacting flows with temperature-dependent viscosity provided by Pember et al.[140] and Day and Bell[141], among others. It is based on the underlying AMReX structure, and is also freely available online. Due to the many capabilities which aligned with our needs, and the great benefit of basing our work on existing software, IAMR was chosen as the foundation for our work. As such, we here present the original algorithm as it was used prior to our alterations.

In IAMR, an advection-diffusion step is used to advance the velocity in time; the solution is then (approximately) projected onto the space of divergence-free fields. The motivation for an approximate rather than exact projection is covered in detail in the literature[142–144]. In addition to solving the Navier-Stokes equations for velocity and pressure, the IAMR code allows for the (conservative or passive) advection of any number of scalar quantities.

The governing equations are given by the variable-density incompressible Navier-Stokes equations, i.e. (2.29) with constant $\eta = \mu$, in addition to an advection equation for each additional scalar. With a single scalar quantity λ, the set of PDEs is

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0, \tag{4.4a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = \frac{1}{\rho}\left(-\nabla p + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{f}\right), \tag{4.4b}$$

$$\frac{\partial \lambda}{\partial t} + \boldsymbol{u} \cdot \nabla \lambda = c_\lambda \nabla^2 \lambda + \boldsymbol{f}_\lambda, \tag{4.4c}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{4.4d}$$

Here, c_λ and f_λ are, respectively, the diffusivity coefficient and active sources for λ.

Algorithm

Presently, we will go through the IAMR algorithm step by step. Note that we will be focusing on the case of a single uniform grid, following closely the deliberation given in the original paper on the algorithm[138]. Velocity, density and scalar concentration are defined at cell centres at integer time steps, and are denoted by $\boldsymbol{u}^m_{i,j,k}$, $\rho^m_{i,j,k}$ and $\lambda^m_{i,j,k}$, respectively. The pressure, on the other hand, is specified at cell corners and is staggered in time, so that it is denoted $p^{m+1/2}_{i+1/2,j+1/2,k+1/2}$. When no subscript is given (e.g. ρ^m), we address the array containing values of ρ in all cells at time t^m. Considerations like initialisation and boundary treatments are ignored at present. Instead, we focus on the steps required in order to evolve the solution a single time step. As a reference, the overall workflow within each time step is given in Algorithm 4.1.

Algorithm 4.1: Advance one time step using IAMR code.

```
input  : $\boldsymbol{u}^m$, $\rho^m$, $\lambda^m$, $p^{m-1/2}$
output : $\boldsymbol{u}^{m+1}$, $\rho^{m+1}$, $\lambda^{m+1}$, $p^{m+1/2}$
Compute time step size $\Delta t$;
begin Advection-diffusion step
    begin Predictor step for normal velocity components
        Compute normal velocity derivatives using slope limiter;
        Evaluate approximations of $\nabla p^{m-1/2}$ and $\nabla^2 \boldsymbol{u}^m$;
        Extrapolate and upwind transverse derivative terms;
        Extrapolate normal velocity components to cell faces at $t^{m+1/2}$ using
          Taylor expansion;
    end
    Perform similar predictor step for tangential velocity components, $\rho$ and $\lambda$;
    Upwind extrapolated normal velocities;
    Apply projection to enforce incompressibility of upwinded velocities;
    Define advection velocities $\boldsymbol{u}^{\text{adv}}$ at faces by subtracting projection gradient;
    Upwind (using $\boldsymbol{u}^{\text{adv}}$) for final time-centred values $\tilde{\boldsymbol{u}}^{m+1/2}$, $\tilde{\rho}^{m+1/2}$, $\tilde{\lambda}^{m+1/2}$;
    Calculate conservative update terms using $\boldsymbol{u}^{\text{adv}}$ and $\tilde{\boldsymbol{u}}^{m+1/2}$, $\tilde{\rho}^{m+1/2}$, $\tilde{\lambda}^{m+1/2}$;
    Advance density to $\rho^{m+1}$ using conservative update formula;
    Compute viscous and diffusive fluxes according to spatial discretisation;
    Solve parabolic equation for intermediate velocity $\boldsymbol{u}^*$;
    Solve parabolic equation for concentration $\lambda^{m+1}$;
end
begin Pressure correction step
    Enforce divergence-free velocity field $\boldsymbol{u}^{m+1}$ by projecting $\boldsymbol{u}^*$;
    Advance pressure to $p^{m+1/2}$;
end
```

The first part of the advection-diffusion step is to extrapolate normal velocity components to each cell face at time $t^{m+1/2}$. This is done through a second-order Taylor series expansion. Consider the cell centre (i, j, k), and suppose we want to extrapolate the first velocity component u to the face $(i + 1/2, j, k)$. Then the truncated Taylor series is

$$u_{i+\frac{1}{2},j,k}^{m+1/2} \approx u_{i,j,k}^{m} + \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_{i,j,k}^{m} + \frac{\Delta t}{2}\left(\frac{\partial u}{\partial t}\right)_{i,j,k}^{m}. \tag{4.5}$$

Taking the first component of (4.4b), we have

$$\begin{aligned}\left(\frac{\partial u}{\partial t}\right)_{i,j,k}^{m} = &-\left(u\frac{\partial u}{\partial x}\right)_{i,j,k}^{m} - \left(v\frac{\partial u}{\partial y}\right)_{i,j,k}^{m} - \left(w\frac{\partial u}{\partial z}\right)_{i,j,k}^{m} \\ &- \frac{1}{\rho_{i,j,k}^{m}}\left(\frac{\partial p}{\partial x}\right)_{i,j,k}^{m-1/2} + \frac{\mu}{\rho_{i,j,k}^{m}}\left(\nabla^2 u\right)_{i,j,k}^{m} + \frac{f_u}{\rho_{i,j,k}^{m}},\end{aligned} \tag{4.6}$$

which can be substituted into the last term of the Taylor expansion to avoid any dependence on temporal derivatives. In order to compute these extrapolated values, however, it is necessary to calculate discrete approximations of the spatial derivatives. This is done separately for each part. The first derivative of normal velocity is evaluated using a monotonicity-limited fourth-order slope approximation S as introduced by Colella in 1985[145]. Since the pressure field is defined at grid nodes, cell-centred approximations of the pressure gradient $\boldsymbol{G}$ are computed by finite differences. The Laplacian is estimated by a second-order finite difference operator L. Finally, transverse derivative terms such as $v\partial u/\partial y$ are evaluated by a separate extrapolation and subsequent upwinding procedure,

$$\left(v\frac{\partial u}{\partial y}\right)_{i,j,k}^{m} \approx \left(\widehat{v\frac{\partial u}{\partial y}}\right)_{i,j,k}^{m}. \tag{4.7}$$

With these discrete representations in place, the intermediate normal velocity at face $i + 1/2$ can be approximated based on the cell to its left (i, j, k) as

$$\begin{aligned}u_{i+1/2,j,k}^{m+1/2,L} = u_{i,j,k}^{m} &+ \left(\frac{\Delta x}{2} - u_{i,j,k}^{m}\frac{\Delta t}{2}\right)(Su)_{i,j,k}^{m} - \frac{\Delta t}{2}\left(\widehat{v\frac{\partial u}{\partial y}}\right)_{i,j,k}^{m} - \frac{\Delta t}{2}\left(\widehat{w\frac{\partial u}{\partial z}}\right)_{i,j,k}^{m} \\ &- \frac{\Delta t}{2\rho_{i,j,k}^{m}}(G_x p)_{i,j,k}^{m-1/2} + \frac{\mu\Delta t}{2\rho_{i,j,k}^{m}}(Lu)_{i,j,k}^{m} + \frac{\Delta t f_u}{2\rho_{i,j,k}^{m}}.\end{aligned} \tag{4.8a}$$

Similarly, an approximation based on the state to its right $(i + 1, j, k)$ is

$$\begin{aligned}u_{i+1/2,j,k}^{m+1/2,R} = u_{i+1,j,k}^{m} &- \left(\frac{\Delta x}{2} + u_{i+1,j,k}^{m}\frac{\Delta t}{2}\right)(Su)_{i+1,j,k}^{m} - \frac{\Delta t}{2}\left(\widehat{v\frac{\partial u}{\partial y}}\right)_{i+1,j,k}^{m} - \frac{\Delta t}{2}\left(\widehat{w\frac{\partial u}{\partial z}}\right)_{i+1,j,k}^{m} \\ &- \frac{\Delta t}{2\rho_{i,j,k}^{m}}(G_x p)_{i+1,j,k}^{m-1/2} + \frac{\mu\Delta t}{2\rho_{i,j,k}^{m}}(Lu)_{i+1,j,k}^{m} + \frac{\Delta t f_u}{2\rho_{i,j,k}^{m}}.\end{aligned} \tag{4.8b}$$

Similar extrapolation procedures must be performed for tangential velocities, density and concentration, except that the latter two require the use of (4.4a) and (4.4c), respectively, instead of (4.4b).

At each face we now have two extrapolated approximations for $\rho^{m+1/2}$, $\lambda^{m+1/2}$ and $\boldsymbol{u}^{m+1/2}$, one from each cell adjoining at the face. In order to pick the most accurate of these, we construct a face-based advection velocity $\boldsymbol{u}^{\text{adv}}$ for upwinding. The final, upwinded, extrapolated approximations at each face are denoted $\tilde{\rho}^{m+1/2}$, $\tilde{\lambda}^{m+1/2}$ and $\tilde{\boldsymbol{u}}^{m+1/2}$. For example, the density at face $(i+1/2, j, k)$, which is located between left (L) and right (R) states, is upwinded according to

$$\tilde{\rho}^{m+1/2}_{i+1/2,j,k} = \begin{cases} \tilde{\rho}^{L}_{i+1/2,j,k} & \text{if } (u^{\text{adv}})_{i+1/2,j,k} > 0, \\ \frac{1}{2}\left(\tilde{\rho}^{L}_{i+1/2,j,k} + \tilde{\rho}^{R}_{i+1/2,j,k}\right) & \text{if } (u^{\text{adv}})_{i+1/2,j,k} = 0, \\ \tilde{\rho}^{R}_{i+1/2,j,k} & \text{if } (u^{\text{adv}})_{i+1/2,j,k} < 0. \end{cases} \tag{4.9}$$

At this point, it is convenient to explain the temporal discretisation employed for the solution of our governing PDEs. In the IAMR code, (4.4a)-(4.4c) are discretised as

$$\frac{\rho^{m+1}-\rho^{m}}{\Delta t} = -[\nabla\cdot(\rho\boldsymbol{u})]^{m+1/2}, \tag{4.10a}$$

$$\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{m}}{\Delta t} = -[\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u})]^{m+1/2} + \frac{1}{\rho^{m+1/2}}\left(-\nabla p^{m-1/2} + \frac{\mu}{2}\left(\nabla^2\boldsymbol{u}^{m} + \nabla^2\boldsymbol{u}^{*}\right) + \boldsymbol{f}\right), \tag{4.10b}$$

$$\frac{\lambda^{m+1}-\lambda^{m}}{\Delta t} = -[\nabla\cdot(\lambda\boldsymbol{u})]^{m+1/2} + \frac{c_\lambda}{2}\left(\nabla^2\lambda^{m} + \nabla^2\lambda^{m+1}\right) + f_\lambda. \tag{4.10c}$$

Here, $\boldsymbol{u}^{*}$ is a provisional new-time velocity field, which does not in general satisfy (4.4d), and which will therefore be updated in the final projection step. The density is centred in time by averaging values at times t^{m} and t^{m+1}. The first term on the right hand side of each equation is a conservative update term, which has been rewritten in a manner which allows us to view it as an advective flux. In the computation of these terms, we utilise the advection velocities and extrapolated approximations as defined above and exemplified by (4.9). The three advective fluxes are $\boldsymbol{F}^{m+1/2}_{\rho} = \tilde{\rho}^{m+1/2}\boldsymbol{u}^{\text{adv}}$, $\boldsymbol{F}^{m+1/2}_{\boldsymbol{u}} = \tilde{\boldsymbol{u}}^{m+1/2}\otimes\boldsymbol{u}^{\text{adv}}$ and $\boldsymbol{F}^{m+1/2}_{\lambda} = \tilde{\lambda}^{m+1/2}\boldsymbol{u}^{\text{adv}}$, and the conservative update terms are thus approximated according to

$$[\nabla\cdot\boldsymbol{F}]^{m+1/2}_{i,j,k} \approx D\left(\boldsymbol{F}^{m+1/2}\right), \tag{4.11}$$

where D is a discrete divergence operator. We can then advance the density in time by rearranging (4.10a) to read

$$\rho^{m+1} = \rho^m - \Delta t D\left(\boldsymbol{F}_\rho^{m+1/2}\right). \tag{4.12}$$

For the viscous and diffuse terms in (4.10b) and (4.10c), the same numerical operator for the Laplacian as defined previously, L, is used. Recalling also the gradient operator $\boldsymbol{G}$, the parabolic equations to be solved numerically for momentum and scalar at the next time step are

$$\boldsymbol{u}^* - \frac{\mu\Delta t}{2\rho^{m+1/2}}L(\boldsymbol{u}^*) = \boldsymbol{u}^m - \Delta t D\left(\boldsymbol{F}_{\boldsymbol{u}}^{m+1/2}\right) + \frac{\Delta t}{\rho^{m+1/2}}\left(-\boldsymbol{G}\left(p^{m-1/2}\right) + \frac{\mu}{2}L(\boldsymbol{u}^m) + \boldsymbol{f}\right), \tag{4.13a}$$

$$\lambda^{m+1} - \frac{c_\lambda\Delta t}{2}L(\lambda^{m+1}) = \lambda^m - \Delta t D\left(\boldsymbol{F}_\rho^{m+1/2}\right) + \frac{c_\lambda\Delta t}{2}L(\lambda^m) + \Delta t f_\lambda. \tag{4.13b}$$

Note that all velocity components are solved for simultaneously.

The final part of the algorithm is the projection step and subsequent pressure update. The intermediate velocity field $\boldsymbol{u}^*$ does not in general satisfy the divergence constraint as given by (4.4d), and it is this which we now seek to rectify. We solve

$$L_\rho\phi = D\left(\frac{1}{\Delta t}\boldsymbol{u}^* + \frac{1}{\rho^{m+1}}\boldsymbol{G}p^{m-1/2}\right), \tag{4.14}$$

where L_ρ is a second-order accurate approximation to $\nabla\cdot\frac{1}{\rho^{m+1}}\nabla$. The new-time velocity is then defined by

$$\boldsymbol{u}^{m+1} = \boldsymbol{u}^* - \Delta t\frac{1}{\rho^{m+1}}\boldsymbol{G}\phi \tag{4.15}$$

and the updated pressure by

$$p^{m+1/2} = \phi. \tag{4.16}$$

The resulting approximate projection satisfies the divergence constraint to second-order accuracy and ensures stability for the algorithm. Given suitable initial and boundary conditions, this algorithm can be used to evolve the system in time.

Time step size

As mentioned in chapter 3, it is essential to use a time step size which provides numerical stability and accurate results for the temporal advancement scheme. Essentially, we wish to find the largest permissible Δt. The investigation of this question by Courant, Friedrichs and Lewy in 1928 for the advection equation[146] is one of the foundations on which the field of numerical solutions for PDEs was built. Consequently, time step restrictions of this type are referred to as CFL conditions. Since IAMR facilitates viscous flow and external body forces in addition to advection, terms relating to these phenomena must also be taken into consideration[147;148]. In IAMR, the thorough derivation by Kang et al.[149] is utilised, which arrives at a time step given by

$$\Delta t = \frac{C_{CFL}}{\frac{1}{2}\left(C_C + C_V + \sqrt{(C_C + C_V)^2 + 4C_F^2}\right)}. \tag{4.17}$$

Here, $C_{CFL} \leq 1$ is a parameter called the CFL coefficient. It is set to 0.9 by default, but can be reduced to smaller values for challenging flow problems. The other coefficients, C_C, C_V and C_F, are constraint coefficients due to convective, viscous and forcing terms, respectively. The convective coefficient is given by

$$C_C = \max_{\Omega}\left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z}\right), \tag{4.18}$$

the viscous coefficient is

$$C_V = 2\max_{\Omega}\left(\frac{\mu}{\rho}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)\right), \tag{4.19}$$

and the coefficient due to external forcing is

$$C_F = \sqrt{\frac{|f_u|}{\Delta x} + \frac{|f_v|}{\Delta y} + \frac{|f_w|}{\Delta z}}. \tag{4.20}$$

4.4 Verification and validation

Although IAMR is a fully functional incompressible flow solver implemented by a range of developers over the years, it was vital for us to design and simulate a range of test cases for verification and validation purposes. In addition to becoming properly convinced that the code solves (4.4) with the desired order of accuracy, the exercise allowed us to become acquainted with usage and development of the software, which was of vital importance for the remainder of the project.

Spatio-temporal convergence study: Taylor-Green vortices

In order to demonstrate second order spatio-temporal convergence of the presented algorithm, we consider the two-dimensional unsteady Taylor-Green vortices, for which an analytical solution exists. This problem is a well-known test case in computational fluid mechanics, and consists of exponentially decaying vortices due to viscous dissipation. The computational domain is $\Omega = [0, \mathcal{L}]^2 \times [0, \mathcal{W}]$, where $\mathcal{W}$ is the width to which we extend the test case in the out-of-plane dimension. Periodic boundary conditions are applied on all domain boundaries, and we take $\mathcal{U}$ as the maximum of the initial velocity distribution. We obtain dimensionless variables by scaling lengths by $\mathcal{L}$, velocity components by $\mathcal{U}$, time by $\mathcal{L}/\mathcal{U}$ and pressure by the kinetic energy density $\rho\mathcal{U}^2$. The Taylor-Green solution is then

$$\hat{u} = \sin(2\pi\hat{x}) \cos(2\pi\hat{y})\, e^{-8\pi^2\hat{t}/Re}, \tag{4.21a}$$

$$\hat{v} = -\cos(2\pi\hat{x}) \sin(2\pi\hat{y})\, e^{-8\pi^2\hat{t}/Re}, \tag{4.21b}$$

$$\hat{p} = \frac{1}{4}\left(\cos(4\pi\hat{x}) + \cos(4\pi\hat{y})\right) e^{-16\pi^2\hat{t}/Re}, \tag{4.21c}$$

where we have introduced the Reynolds number

$$Re = \frac{\rho\mathcal{U}\mathcal{L}}{\mu}. \tag{4.22}$$

For a series of spatial resolutions, characterised by the amount of cells N in the x- and y-directions, our system is initialised to the distribution given by inserting $\hat{t} = 0$ into (4.21) with $Re = 100$, and advanced to the time $\hat{t} = 1$. The resulting velocity field is subtracted from the analytical solution in each cell, and we denote this residual $\varepsilon_N(\boldsymbol{x})$. For two meshes with resolution N and $2N$, the convergence rate of our numerical method can be computed as

$$r_* = \log_2 \left\| \frac{\varepsilon_N}{\varepsilon_{2N}} \right\|_* \tag{4.23}$$

where $\|\cdot\|_*$ is an appropriate function norm, typically one of

$$\|\varepsilon\|_1 = \int_\Omega |\varepsilon(\boldsymbol{x})| \,\mathrm{d}V, \tag{4.24a}$$

$$\|\varepsilon\|_2 = \left(\int_\Omega |\varepsilon(\boldsymbol{x})|^2 \,\mathrm{d}V\right)^{\frac{1}{2}}, \tag{4.24b}$$

$$\|\varepsilon\|_\infty = \max_\Omega |\varepsilon(\boldsymbol{x})|. \tag{4.24c}$$

Table 4.1 shows how the convergence rate for our solver approaches two in the discrete versions of all these norms as N grows, as expected.

N	$\|\varepsilon\|_1$	r_1	$\|\varepsilon\|_2$	r_2	$\|\varepsilon\|_\infty$	r_∞
32	4.46×10^{-4}	–	5.68×10^{-4}	–	1.28×10^{-3}	–
64	1.84×10^{-4}	1.28	2.30×10^{-4}	1.31	4.73×10^{-4}	1.44
128	5.22×10^{-5}	1.82	6.48×10^{-5}	1.83	1.31×10^{-4}	1.86
256	1.36×10^{-5}	1.94	1.68×10^{-5}	1.95	3.37×10^{-5}	1.95
512	3.45×10^{-6}	1.98	4.26×10^{-6}	1.98	8.55×10^{-6}	1.98

Table 4.1: Error norms and convergence rates for viscous incompressible flow in the Taylor-Green vortex.

Rayleigh problem

The Rayleigh problem[150], also known as Stokes' first problem, is one of the simplest applications of the Navier-Stokes equations which has an exact analytical solution for the unsteady case. It consists of the temporal evolution of an infinite incompressible shear layer, initiated by the instantaneous start-up movement of a plate. We use a finite computational domain equal to $\Omega = [0, \mathcal{L}]^2$, and simulate opposing plates, taking the initial velocity profile as

$$v(x,0) = \begin{cases} -\mathcal{U}, & x < 0.5\mathcal{L} \\ \mathcal{U}, & x \geq 0.5\mathcal{L} \end{cases}. \tag{4.25}$$

At the left and right edges, the initial conditions are imposed, while we employ periodic boundary conditions in the y-direction. The characteristic speed is of course $\mathcal{U}$, and we scale lengths by $\mathcal{L}$, although the problem doesn't have an inherent characteristic length scale. Times are made dimensionless through division by $\rho\mathcal{L}^2/(4\mu)$, yielding an analytical solution given by

$$\hat{v}(\hat{x},\hat{t}) = \operatorname{erf}\left(\frac{\hat{x}}{\sqrt{\hat{t}}}\right), \tag{4.26}$$

where $\operatorname{erf}(x)$ is the Gauss error function. Results are shown in Figure 4.4 for $N = 32$ cells at three different times. They overlap nicely with the analytical solution as time goes by.

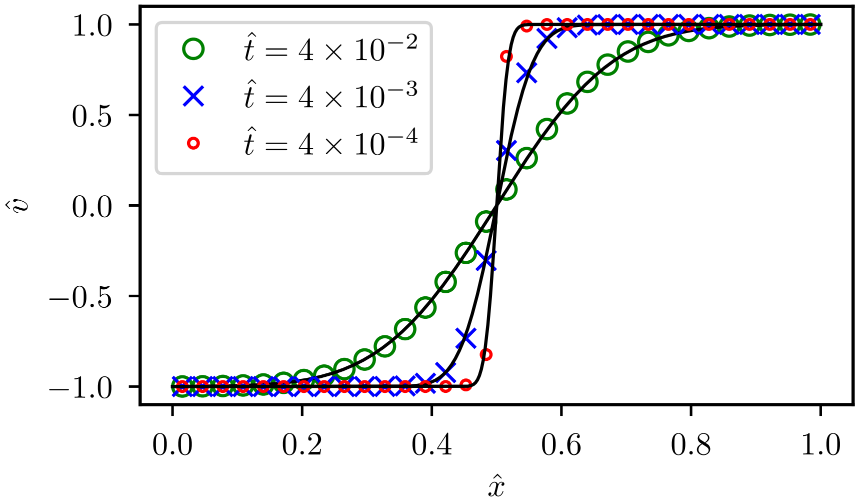


Figure 4.4: Numerical and analytical (solid lines) velocity profiles for increasing time in the Rayleigh problem.

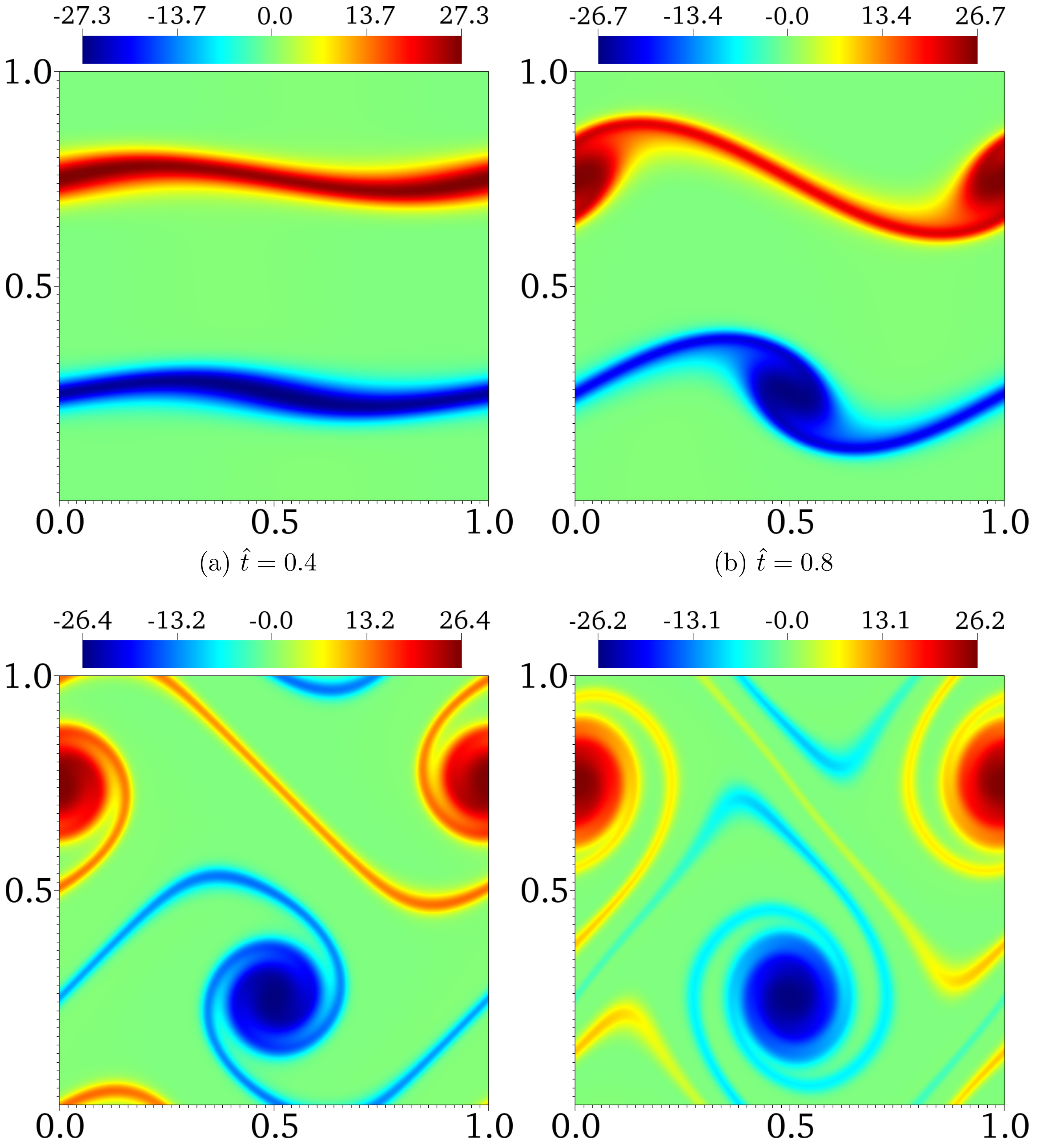
Double shear layer

Bell, Colella and Glaz introduced a test case for the unsteady, incompressible Navier-Stokes equations in 1989[151], which contains a double shear layer with a large initial velocity gradient. Periodic boundary conditions are enforced in both directions of the domain $\Omega = [0, \mathcal{L}]^2$, and the velocity field is initially given by a horizontal shear layer which is perturbed by a small-amplitude vertical velocity, according to

$$
\begin{aligned}
u(x, y, 0) &= \mathcal{U} \times \begin{cases} \tanh(30(y - \mathcal{L}/4)), & y \leq \mathcal{L}/2 \\ \tanh(30(3\mathcal{L}/4 - y)), & y > \mathcal{L}/2 \end{cases}, \\
v(x, y, 0) &= 0.05\,\mathcal{U} \sin(2\pi x/\mathcal{L}).
\end{aligned}
$$

The characteristic length and speed scales are $\mathcal{L}$ and $\mathcal{U}$, respectively.

Results which depict the vorticity field, for the case with $Re = 5000$, are shown in figure 4.5. The results clearly show the evolution of the initial shear layers to a periodically recurring pattern of large eddies, exactly as expected. When comparing the color plots in figure 4.5 with the original contour plots produced by Bell, Colella and Glaz[151], or with solutions computed with modern techniques as done by Dumbser et al.[152], it is clear that our solver captures the same physics with high accuracy.



(a) $\hat{t} = 0.4$

(b) $\hat{t} = 0.8$

(c) $\hat{t} = 1.2$

(d) $\hat{t} = 1.6$

Figure 4.5: Temporal evolution of the double shear layer with $N = 256$. The color plot depicts the vorticity scaled by the characteristic strain rate $\mathcal{U}/\mathcal{L}$ after (a) 0.4, (b) 0.8, (c) 1.2 and (d) 1.6 dimensionless time units.

Couette-Poiseuille flow

So far, all the validation test cases have utilised periodic boundary conditions. In order to ensure that the code can handle no-slip wall conditions, we consider the two simplest flow problems involving them. Planar Couette flow is characterised by a channel where the top and bottom walls have prescribed constant velocity values in the transverse direction, which induces viscous shearing in the region between them. If these are the only forces driving the motion, the resulting velocity field is merely a linear function between the wall values. Poiseuille flow, on the other hand, is driven by a constant applied pressure gradient aligned with the channel, and is therefore also simply referred to as pressure-driven flow. Its analytical solution is a parabolic profile over the channel width. For Newtonian fluids, there is a general analytical solution for combined Couette-Poiseuille flow, i.e. problems with a combination of viscous shear layers due to moving walls and an applied pressure gradient.

The channel is bounded by walls at $y = 0$ and $y = \mathcal{W}$, and is infinite in the other spatial directions. On these walls, we prescribe the speeds $\mathcal{U}_0$ and $\mathcal{U}_\mathcal{W}$, respectively. In the direction of flow, we apply the constant pressure gradient $\mathcal{P}/\mathcal{W}$. Scaling lengths by $\mathcal{W}$ and velocities by $\mathcal{U}_\mathcal{W} > 0$, the analytical solution is given by

$$\hat{u}(\hat{y}) = \frac{\mathcal{U}_0}{\mathcal{U}_\mathcal{W}} + \left(1 + Po - \frac{\mathcal{U}_0}{\mathcal{U}_\mathcal{W}}\right) \hat{y} - Po\, \hat{y}^2, \tag{4.27}$$

where the Poiseuille number

$$Po = \frac{\mathcal{P}\mathcal{W}}{2\mu\mathcal{U}_\mathcal{W}} \tag{4.28}$$

is a dimensionless group quantifying the ratio of pressure gradient to viscous forces. By moving the problem to the frame of reference moving with the lower wall, we can set $\mathcal{U}_0 = 0$, ending up with the Poiseuille number as our only free variable:

$$\hat{u}(\hat{y}) = (1 + Po)\, \hat{y} - Po\, \hat{y}^2. \tag{4.29}$$

In figure 4.6, we show that our computed results with $Po = 4$ overlap nicely with the analytical solution given by (4.29) without having to use a high spatial resolution at all. The numerical implementation of the no-slip boundary condition with prescribed wall speed is therefore valid.

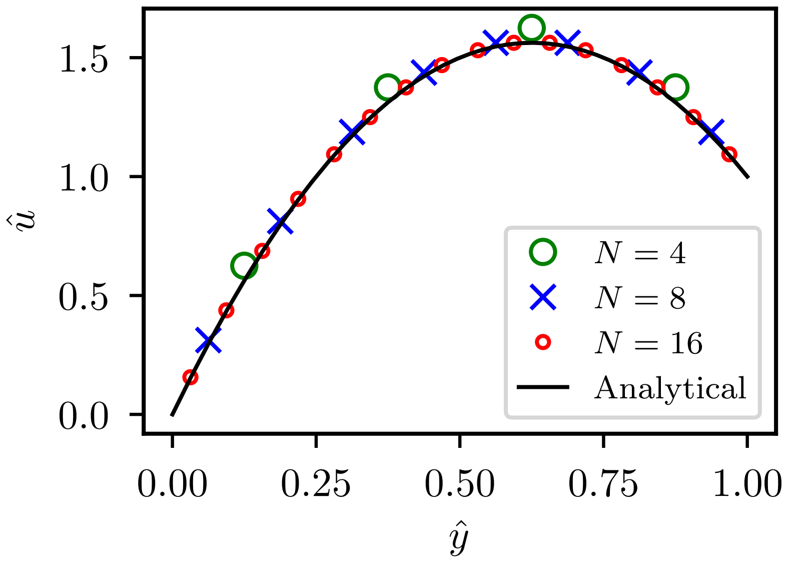


Figure 4.6: Comparison of our numerical results with the analytical velocity profile for Couette-Poiseuille flow. Here, N denotes the number of computational cells across the gap width $\mathcal{W}$.

Parallel performance

In order to demonstrate how well the parallel processing works in IAMR, we performed a weak scaling study on the supercomputer Cori[1]. This was set up as a replication problem, where a base test is reproduced with higher and higher core counts. In the base test, a three-dimensional unit cube with periodic boundary conditions is initialised with constant density and viscosity, and unidirectional velocity field. Within this system, a passive scalar is advected, and the computational runtime per simulation time step is taken as the base reference. We then double the domain length (and number of cells) in each direction, leading to eight unit cubes, within each of which the same problem is set up. This ensures that the same spatial discretisation and time step size is used although the problem is much larger. Eight times as many cores are used to solve the problem. As seen from the results in figure 4.7, we achieve excellent performance, with only a slight increase in runtime per time step when moving from a couple of million cells to over a billion.

[1] https://www.nersc.gov/users/computational-systems/cori/

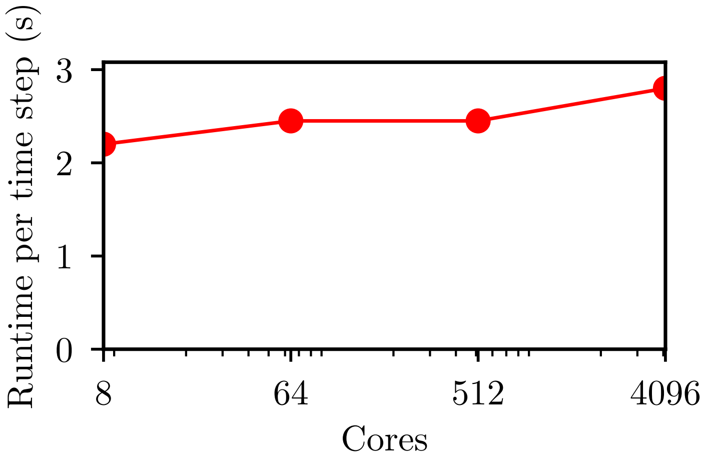


Figure 4.7: Results from replication study for weak scaling in the IAMR code. Note that the primary axis utilises logarithmic scaling. Ideal parallelisation would result in a straight horizontal line. It is evident that massive parallelisation on clusters such as Cori can be utilised to simulate large problems in IAMR.

Chapter 5

Large-scale simulations of yield stress fluids

Numerical treatment of viscoplastic flow problems has significantly improved over the last decades. Researchers interested in analysing such flows are, however, still limited by the computational cost associated with their solution. Notably, most research published in the field considers only two spatial dimensions and only steady-state solutions. Although open-source libraries such as FEniCS[118] and OpenFOAM[116] can be used to simulate regularised viscoplastic fluids in three dimensions, our contribution aims to provide a massively parallelisable tool for contemporary supercomputer architectures, with capabilities for structured AMR. In order to achieve this, we begin with the IAMR[138] code, that uses a second-order accurate, approximate projection method to solve the incompressible Navier-Stokes equations. We have implemented the Papanastasiou regularised Herschel-Bulkley model for the apparent viscosity function in IAMR so that it can be used to simulate generalised Newtonian fluids. As such, we are able to take advantage of modern supercomputer architectures in our simulation of viscoplastic fluids. For the first time, results are presented for the transient, three-dimensional lid-driven cavity problem for viscoplastic fluids. Notably, the evolution of the three-dimensional yield surface from rest to steady-state is tracked, before the lid velocity is set to zero, allowing cessation of the viscoplastic fluid.

Our motivation for this approach was the potential to greatly increase the speed of viscoplastic flow simulations, using methods which have never been tried previously. According to lecture notes given by Anthony Wachs on computational methods for viscoplastic fluid flows[153], "*structured adaptive mesh refinement has never been tried for viscoplastic flows, although it should perform quite nicely. [...] there has been* *no report so far in the literature on the use of AMR (regardless of unstructured or structured) for the solution of time-dependent viscoplastic flows.*" Since the methods we are interested in developing employ structured AMR for unsteady flow, there was a great opportunity for making an impact in the field of computational viscoplasticity.

5.1 Extending IAMR to generalised Newtonians

In the following, we will show how a code which solves (4.4) must be altered when the fluid is a generalised non-Newtonian. The question is then whether the IAMR code can be augmented to allow simulation of these types of fluids while at the same time keeping a high-order convergence rate and retaining the benefits of massive parallelisation. Additionally, we want to make sure that any further time step constraints remain acceptable. Since we want to solve the more general (2.29), we need to make the viscosity dependent on the strain rate. We replace the viscous term in (4.4b), so that

$$\mu\nabla^2\boldsymbol{u} \rightarrow \nabla\cdot(\eta\dot{\boldsymbol{\gamma}}) = \eta\left(\nabla\cdot\dot{\boldsymbol{\gamma}}\right) + (\nabla\eta)\cdot\dot{\boldsymbol{\gamma}} = \eta\nabla^2\boldsymbol{u} + (\nabla\eta)\cdot\dot{\boldsymbol{\gamma}}$$

The momentum balance equation thus becomes

$$\rho\left(\frac{\partial\boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \eta\nabla^2\boldsymbol{u} + (\nabla\eta)\cdot\dot{\boldsymbol{\gamma}} + \boldsymbol{f}, \tag{5.1}$$

$$\tag{5.2}$$

Comparing the above equation with (4.4b), the only differences are the replacement of μ by η in the Laplacian term and the addition of a term which takes into consideration the gradient of the viscosity.

Algorithm

We will presently revisit algorithm 4.1, while focusing on the steps where alterations need to be made for nonlinear rheological equations. To start with, we consider the predictor step for normal velocity components, i.e. code lines 2-7 in the algorithm. Since the only difference between the two momentum equations is the replacement of μ by the apparent viscosity function η, there are two main alterations that need to be made. For the cell (i,j,k) and face $(i+\frac{1}{2},j,k)$, we must first compute the apparent viscosity $\eta^m_{i,j,k}$ using (3.24) and suitable approximations of the velocity gradient $\nabla\boldsymbol{u}$.

Subsequently, we need to alter the extrapolation defined in (4.8) to also consider the velocity gradient terms.

The reasoning why we need to compute approximations of the components of $\nabla \boldsymbol{u}$ is as follows. From (3.24), we know that the apparent viscosity function depends on the rate-of-strain magnitude alone, i.e. $\eta = \eta(\|\dot{\boldsymbol{\gamma}}\|)$. The tensor can be written out in full as

$$\dot{\boldsymbol{\gamma}} = \nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\top = \begin{bmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2\frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} & 2\frac{\partial 2}{\partial z} \end{bmatrix}, \tag{5.3}$$

By using the definition of the tensor norm as given in (2.3), an explicit formula for the magnitude of the rate-of-strain tensor, which depends only on components of the velocity gradient, is given by

$$\|\dot{\boldsymbol{\gamma}}\| = \sqrt{2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + 2\left(\frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)^2}. \tag{5.4}$$

We therefore require cell-centred spatial derivatives of all velocity components in each direction, allowing us to compute $\dot{\gamma}^m_{i,j,k}$, and, consequently, $\eta^m_{i,j,k}$. The spatial derivatives are straight-forward to calculate using finite differences based on the available computed velocity values.

Once the viscosity distribution has been computed, we then need to approximate the gradient of this field again by finite differences, before finally altering (4.8). The gradient of the apparent viscosity field is necessary in order to evaluate the term

$$\nabla \eta \cdot \dot{\boldsymbol{\gamma}} = \begin{bmatrix} 2\frac{\partial \eta}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial \eta}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{\partial \eta}{\partial z}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \frac{\partial \eta}{\partial x}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + 2\frac{\partial \eta}{\partial y}\frac{\partial v}{\partial y} + \frac{\partial \eta}{\partial z}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \frac{\partial \eta}{\partial x}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) + \frac{\partial \eta}{\partial y}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) + 2\frac{\partial \eta}{\partial z}\frac{\partial w}{\partial z} \end{bmatrix}. \tag{5.5}$$

With these explicit formulas for computing the stress divergence

$$\nabla \cdot \boldsymbol{\tau} = \eta \; \nabla^2 \boldsymbol{u} + \nabla \eta \cdot \dot{\boldsymbol{\gamma}}, \tag{5.6}$$

we denote by ∂_x the finite difference approximation of its x-component, i.e. the numerical estimate of

$$(\nabla \cdot \boldsymbol{\tau}) \cdot \hat{\boldsymbol{e}}_x = \eta \nabla^2 u + 2\frac{\partial \eta}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial \eta}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{\partial \eta}{\partial z}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right). \tag{5.7}$$

Note that with the exception of the Laplacian term, in which μ has been replaced by η^m, all new additions to the extrapolation formulas contain components of the viscosity gradient. The modified versions of (4.8) are then given by

$$\begin{aligned} u^{m+1/2,L}_{i+1/2,j,k} = u^m_{i,j,k} + \left(\frac{\Delta x}{2} - u^m_{i,j,k}\frac{\Delta t}{2}\right)(Su)^m_{i,j,k} - \frac{\Delta t}{2}\left(v\widehat{\frac{\partial u}{\partial y}}\right)^m_{i,j,k} - \frac{\Delta t}{2}\left(w\widehat{\frac{\partial u}{\partial z}}\right)^m_{i,j,k} \\ + \frac{\Delta t}{2\rho^m_{i,j,k}}\left(-\left(G_x p\right)^{m-1/2}_{i,j,k} + \left(\partial_x \boldsymbol{\tau}\right)^m_{i,j,k} + f_u\right). \end{aligned} \tag{5.8a}$$

and

$$\begin{aligned} u^{m+1/2,R}_{i+1/2,j,k} = u^m_{i+1,j,k} - \left(\frac{\Delta x}{2} + u^m_{i+1,j,k}\frac{\Delta t}{2}\right)(Su)^m_{i+1,j,k} - \frac{\Delta t}{2}\left(v\widehat{\frac{\partial u}{\partial y}}\right)^m_{i+1,j,k} - \frac{\Delta t}{2}\left(w\widehat{\frac{\partial u}{\partial z}}\right)^m_{i+1,j,k} \\ + \frac{\Delta t}{2\rho^m_{i+1,j,k}}\left(-(G_x p)^{m-1/2}_{i+1,j,k} + \left(\partial_x \boldsymbol{\tau}\right)^m_{i+1,j,k} + f_u\right). \end{aligned} \tag{5.8b}$$

Similar modifications are necessary for the other velocity components. With a plan for discretisation and extrapolation in place, lines 1-8 in algorithm 4.1 can be augmented. Although we are only considering the first component of the momentum balance equation, and only describing the extrapolation of a single velocity component to one face, further generalisations follow naturally. In summary, our modified algorithm should, at each cell centre (i, j, k):

1. Approximate the velocity gradients using finite differences.
2. Compute the magnitude of the rate-of-strain tensor according to (5.4).
3. Evaluate the viscosity as a function of the strain rate magnitude, $\eta^m_{i,j,k} = \eta\left(\|\dot{\boldsymbol{\gamma}}\|^m_{i,j,k}\right)$, using one of the rheological equations in section 2.2.
4. Approximate the viscosity gradient using finite differences.
5. Extrapolate variables according to (5.8).

The steps on lines 9-14 do not require any changes for viscoplastic flow, so we continue to the computation of viscous fluxes, discretisation in time and parabolic solves. When replacing (4.4b) by (2.29b), the temporal discretisation must be expanded to include the additional terms arising from viscosity gradients. We replace (4.10b) and (4.13a) by

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^m}{\Delta t} = -[\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u})]^{m+1/2} + \frac{1}{\rho^{m+1/2}}\left(-\nabla p^{m-1/2} + \frac{1}{2}\left(\nabla\cdot\boldsymbol{\tau}\left(\boldsymbol{u}^m\right) + \nabla\cdot\boldsymbol{\tau}\left(\boldsymbol{u}^*\right)\right) + \boldsymbol{f}\right) \tag{5.9}$$

and

$$\boldsymbol{u}^* - \frac{\Delta t}{2\rho^{m+1/2}}\boldsymbol{\partial\tau}(\boldsymbol{u}^*) = \boldsymbol{u}^m - \Delta t D\left(\boldsymbol{F}_{\boldsymbol{u}}^{m+1/2}\right) + \frac{\Delta t}{\rho^{m+1/2}}\left(-\boldsymbol{G}\left(p^{m-1/2}\right) + \frac{1}{2}\boldsymbol{\partial\tau}(\boldsymbol{u}^m) + \boldsymbol{f}\right). \tag{5.10}$$

It is therefore necessary to build a right hand side for the parabolic solve which depends on the viscosity field. In addition to this, all the viscosity gradient terms as given in (5.5) must be included. Conceptually, these steps do not encompass many changes, but there are many small computational considerations involved in expanding the momentum update equations. Note that the numerical approximation of the stress tensor utilises η^m regardless of the input velocity, due to the nonlinear dependency an implicit viscosity would introduce in the parabolic solver. Consequently, the numerical integration in time is not guaranteed to be of second order. As seen later in this chapter, however, our results match the time-stepping results from literature accurately. This drawback is alleviated in the reworked algorithm presented in chapter 7, where the conservative form of the stress divergence is treated using a second-order Runge-Kutta scheme.

Handling interfaces between refinement levels

We need to ensure that the dynamic meshing capabilities also work with variable viscosity. The subcyling procedure with synchronization, as described in section 4.2, and illustrated in figure 4.3, is performed for the primary variables. Since this includes the velocity field, whose gradient the apparent viscosity depends on, the latter is also in sync as long as the finer grids have filled boundary cells so that finite differences can be computed. But this is guaranteed by the fact that the coarse grid patches always contain one layer of cells which surround the finer, properly nested patches of cells. When a new refinement patch is created, the underlying coarse grid data is interpolated in order to initialise data at the finer level.

Computation of time step size

Although the original IAMR code utilises a CFL criterion when computing the time step, it doesn't allow variable viscosity, so μ must be replaced by the apparent viscosity, η in (4.19). However, direct substitution results in a computed Δt which is overly restrictive for viscoplastic flows where the Bingham number is large. Since such flows account for many of the scenarios we are interested in being able to handle, we make an effort to find an alternative viscous CFL coefficient. The problem with direct maximisation of η is that for viscoplastic flows, the apparent viscosity has a singularity in unyielded regions. Even for regularised models, very large viscous terms appear in parts of our domain. On the other hand, Syrakos et al.[64] demonstrated that the time step size for such flows is actually dependent on the ratio of the Reynolds to Bingham numbers. In their article, they simulated cessation of Bingham fluids in the two-dimensional lid-driven cavity, and found that it was sufficient to utilise a time step

$$\Delta t_{\text{Syrakos}} \propto \frac{Re}{Bi+1}\Delta x^2, \tag{5.11}$$

where the proportionality factor is $\mathcal{O}(1)$. Notably, (5.11) does not need to take the level of regularisation into account (they used $Pa = 400$). In fact, inserting the numbers from their study into (4.19) gives the ratio between the two as

$$\frac{Bi+1}{BiPa+1} \approx \frac{1}{Pa}, \tag{5.12}$$

where the approximation holds for large $Bi, Pa >> 1$. Consequently, the time step is much smaller than it needs to be, by a factor inversely proportional to the Papanastasiou number. We therefore remove the dependency of (4.19) on the regularisation parameter, and instead replace it by

$$C_V = \frac{2\mu(1+Bi)}{\rho} \max_{\Omega} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right). \tag{5.13}$$

for viscoplastic flows.

Test for convergence to steady-state

Many of the test cases employed for testing viscoplastic fluid codes in the literature have steady-state solutions, so it is convenient if we can advance our system in time until such a state is reached. IAMR allows the user to set a stop time and a maximum number of time steps to advance, and will cancel when either of these is reached. Additionally, we have added functionality tests for convergence to steady-state if the user supplies a negative stop time (e.g. $t_{\text{end}} = -1$). We compute the magnitude of the stress tensor at every time step according to

$$\|\boldsymbol{\tau}\|^m = \eta^m \|\dot{\boldsymbol{\gamma}}\|^m . \tag{5.14}$$

Steady-state is signalled as achieved when the largest changes in this magnitude from one iteration to the next becomes smaller than some user-prescribed tolerance, by default set to 10^{-4} following Muravleva and Olshanskii [154].

5.2 Verification: Bingham plastic convergence study

In order to evaluate the convergence of our code for viscoplastic fluids, we consider plane Poiseuille flow between parallel plates, as discussed in section 2.3. Although (2.55) gives us the analytical solution for a Bingham fluid, it would not be correct to use for convergence analysis, since the Papanastasiou regularisation which we are employing is not an equivalent formulation. We have therefore derived the analytical solution for a Papanastasiou regularised Bingham fluid, which has never been done previously.

Recall that plane Poiseuille flow in a channel of width $2\mathcal{W}$ in the y-direction is driven by a constant applied pressure gradient $\mathcal{P}/\mathcal{W}$ in the x-direction, resulting in a steady-state solution with non-zero x-velocity aligned with the channel plates. The velocity profile is only a function of y, and takes its maximum value at the centre of the channel, $y = 0$. The plane $y = 0$ is a symmetry plane for the solution, so we solve for $y \geq 0$. We take $\mathcal{W}$ as the characteristic length, while the characteristic speed $\mathcal{U}$ is the maximum of the unregularised analytical solution. Scaling by these, in addition to strain rate $\mathcal{U}/\mathcal{W}$ and stress $\mu\mathcal{U}/\mathcal{W}$, the steady-state governing equations in dimensionless form are

$$\hat{\nabla}\hat{p} = \hat{\nabla} \cdot \hat{\boldsymbol{\tau}}, \tag{5.15a}$$

$$\hat{p} = -Po\,\hat{x}\ , \tag{5.15b}$$

$$\hat{\boldsymbol{\tau}} = \left(1 + Bi\,\frac{1 - \exp\left(-Pa\,\left\|\hat{\dot{\boldsymbol{\gamma}}}\right\|\right)}{\left\|\hat{\dot{\boldsymbol{\gamma}}}\right\|}\right)\hat{\dot{\boldsymbol{\gamma}}}. \tag{5.15c}$$

Since the only non-zero velocity component is $u = u(y)$, the magnitude of the rate-of-strain tensor is $\|\dot{\boldsymbol{\gamma}}\| = |\partial u/\partial y|$. Furthermore, since u is monotonically decreasing for $y > 0$, we have $|\partial u/\partial y| = -\partial u/\partial y$. With this information in place, the first component of (5.15a) is

$$-\frac{Bi}{y_0} = \frac{\partial}{\partial \hat{y}}\left(\frac{\partial \hat{u}}{\partial \hat{y}}\left(1 - Bi\,\frac{1 - \exp\left(Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}\right)}{\partial \hat{u}/\partial \hat{y}}\right)\right), \tag{5.16}$$

where $y_0 = Bi/Po$ is the location of the yield surface, and the only free parameter in the unregularised Bingham Poiseuille problem. Equation (5.16) is a separable ordinary differential equation for $\partial \hat{u}/\partial \hat{y}$:

$$-\frac{2}{(1-y_0)^2} = \left(1 + \frac{2y_0 Pa}{(1-y_0)^2}\,\exp\left(Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}\right)\right)\frac{\partial}{\partial \hat{y}}\left(\frac{\partial \hat{u}}{\partial \hat{y}}\right), \tag{5.17}$$

For ease of writing, we introduce $\xi = 2y_0 Pa/(1-y_0)^2$. Integrating (5.17) then leads to an implicit solution for $\partial \hat{u}/\partial \hat{y}$, given by

$$\frac{\partial \hat{u}}{\partial \hat{y}} + \frac{\xi}{Pa}\exp\left(Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}\right) = -2\frac{\hat{y} - y_0}{(1-y_0)^2}, \tag{5.18}$$

where we have used the fact that $\partial u/\partial y = 0$ at $y = 0$. In order to obtain an explicit solution, we rewrite (5.18) in the form

$$\xi e^{Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}}\exp\left(\xi e^{Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}}\right) = \xi e^{-\xi(\hat{y}-y_0)/y_0}. \tag{5.19}$$

Equation (5.19) is on a form which facilitates the use of the Lambert W function, defined as the solution to $x = W(xe^x)$, and consequently also referred to as the product logarithm. Using the definition of $W(x)$, we get

$$\xi e^{Pa\,\frac{\partial \hat{u}}{\partial \hat{y}}} = W\left(\xi e^{-\xi(\hat{y}-y_0)/y_0}\right), \tag{5.20}$$

which can be inserted into (5.18) in order to find the explicit solution,

$$\frac{\partial \hat{u}}{\partial \hat{y}} = -2\frac{\hat{y}-y_0}{(1-y_0)^2} - \frac{1}{Pa}W\left(\xi e^{-\xi(\hat{y}-y_0)/y_0}\right). \tag{5.21}$$

Using the no-slip boundary condition $\hat{u}(1) = 0$, and the identity

$$\int W\left(Ae^{Bx}\right)\mathrm{d}x = \frac{1}{2B}\left(1 + W\left(Ae^{Bx}\right)\right)^2 + C, \tag{5.22}$$

direct integration and some manipulation then gives the analytical expression for the velocity profile as

$$\hat{u}(\hat{y}) = 1 - \left(\frac{\hat{y}-y_0}{1-y_0}\right)^2 + \frac{y_0}{2\xi Pa}\left(\left(1 + W\left(\xi e^{-\xi(\hat{y}-y_0)/y_0}\right)\right)^2 - \left(1 + W\left(\xi e^{-\xi(1-y_0)/y_0}\right)\right)^2\right). \tag{5.23}$$

This analytical solution is what we expect our numerical results to converge to, when taking into account the effect of regularisation.

We emphasise that the newly derived analytical solution given by (5.23) takes into account the effect of the regularisation parameter on the analytical solution, so that its influence is removed from the numerical algorithm in our convergence study. We therefore use a moderate $Pa = 100$, which in any case is small enough to capture the yield stress effects on this simple problem, and vary the amounts of grid points $N = 2\mathcal{W}/\Delta x$ over the channel width. The experiment is repeated for several values of y_0, since we expect the condition number of the numerical problem to increase with the Bingham number. As seen in table 5.1, the numerical solution converges to (5.23) with increasing spatial accuracy for all values of y_0. Since the analytical solution is parabolic for the Newtonian case, we achieve a perfect second order convergence rate, but is drops to slightly lower than two as the yield stress increases. This is as expected, and the solution in the unyielded region (which is undefined for Bingham fluids) converges slowest. Following the convergence analysis of Olshanskii[155], table 5.2 shows the corresponding convergence rates when we only consider the yielded region ($\hat{y} > y_0$), which are much better even for $y_0 = 0.5$. The ∞-norm is the same, since it measures the maximum error throughout the domain, which is immediately outside the yield surface in all cases. Although the strongly non-linear viscosity function poses a significant numerical challenge, our algorithm performs well. Even more rapid convergence could possibly be achieved through the use of convergence acceleration methods such as those discussed by Housiadas[156].

y_0	N	$\|\varepsilon\|_1$	r_1	$\|\varepsilon\|_2$	r_2	$\|\varepsilon\|_\infty$	r_∞
0	16	3.91×10^{-3}	–	3.91×10^{-3}	–	3.91×10^{-3}	–
	32	9.77×10^{-4}	2.00	9.77×10^{-4}	2.00	9.77×10^{-4}	2.00
	64	2.44×10^{-4}	2.00	2.44×10^{-4}	2.00	2.44×10^{-4}	2.00
	128	6.10×10^{-5}	2.00	6.10×10^{-5}	2.00	6.10×10^{-5}	2.00
0.1	16	4.05×10^{-3}	–	4.99×10^{-3}	–	1.05×10^{-2}	–
	32	1.23×10^{-3}	1.71	1.69×10^{-3}	1.57	3.85×10^{-3}	1.45
	64	3.51×10^{-4}	1.81	5.26×10^{-4}	1.68	1.31×10^{-3}	1.55
	128	9.56×10^{-5}	1.87	1.52×10^{-4}	1.79	4.13×10^{-4}	1.67
	256	2.51×10^{-5}	1.93	4.14×10^{-5}	1.88	1.18×10^{-4}	1.81
0.2	16	1.02×10^{-2}	–	1.42×10^{-2}	–	2.49×10^{-2}	–
	32	3.33×10^{-3}	1.61	5.01×10^{-3}	1.50	9.45×10^{-3}	1.40
	64	1.01×10^{-3}	1.72	1.61×10^{-3}	1.64	3.18×10^{-3}	1.57
	128	2.92×10^{-4}	1.79	4.81×10^{-4}	1.74	1.03×10^{-3}	1.63
	256	8.06×10^{-5}	1.86	1.36×10^{-4}	1.83	3.12×10^{-4}	1.72
0.5	16	1.22×10^{-1}	–	1.34×10^{-1}	–	1.63×10^{-1}	–
	32	4.20×10^{-2}	1.54	4.80×10^{-2}	1.48	6.17×10^{-2}	1.40
	64	1.38×10^{-2}	1.61	1.63×10^{-2}	1.56	2.24×10^{-2}	1.46
	128	4.38×10^{-3}	1.66	5.34×10^{-3}	1.61	7.64×10^{-3}	1.55
	256	1.34×10^{-3}	1.70	1.67×10^{-3}	1.67	2.55×10^{-3}	1.58

Table 5.1: Error norms and convergence rates for plane Poiseuille flow of a Papanastasiou regularised Bingham fluid with $Pa = 100$.

y_0	N	$\|\varepsilon\|_1$	r_1	$\|\varepsilon\|_2$	r_2	$\|\varepsilon\|_\infty$	r_∞
0.1	16	3.12×10^{-3}	–	3.57×10^{-3}	–	6.50×10^{-3}	–
	32	8.65×10^{-4}	1.85	1.08×10^{-3}	1.72	3.02×10^{-3}	1.11
	64	2.59×10^{-4}	1.74	3.85×10^{-4}	1.50	1.31×10^{-3}	1.20
	128	6.75×10^{-5}	1.94	1.08×10^{-4}	1.83	4.13×10^{-4}	1.67
	256	1.65×10^{-5}	2.03	2.73×10^{-5}	1.99	1.18×10^{-4}	1.81
0.2	16	5.33×10^{-3}	–	8.01×10^{-3}	–	2.49×10^{-2}	–
	32	2.02×10^{-3}	1.40	3.48×10^{-3}	1.20	9.45×10^{-3}	1.40
	64	5.57×10^{-4}	1.86	1.06×10^{-3}	1.72	3.18×10^{-3}	1.57
	128	1.33×10^{-4}	2.06	2.76×10^{-4}	1.94	1.03×10^{-3}	1.63
	256	3.44×10^{-5}	1.95	7.51×10^{-5}	1.88	3.12×10^{-4}	1.72
0.5	16	8.19×10^{-2}	–	9.80×10^{-2}	–	1.58×10^{-1}	–
	32	2.37×10^{-2}	1.79	3.12×10^{-2}	1.65	6.17×10^{-2}	1.36
	64	6.45×10^{-3}	1.88	9.38×10^{-3}	1.73	2.24×10^{-2}	1.46
	128	1.72×10^{-3}	1.91	2.73×10^{-3}	1.78	7.64×10^{-3}	1.55
	256	4.52×10^{-4}	1.93	7.76×10^{-4}	1.81	2.55×10^{-3}	1.58

Table 5.2: Error norms and convergence rates for plane Poiseuille flow of a Papanastasiou regularised Bingham fluid with $Pa = 100$, using only the solution in the yielded region.

5.3 Lid-driven cavity

In order to validate our code, we consider the two-dimensional lid driven cavity test case. It is a historically significant and widely used benchmark problem for viscous flow simulations, and consequently a large amount of reference results exist in the literature. Our domain is a square of side length $\mathcal{L}$, i.e. $\Omega = [0, \mathcal{L}]^2$, and is filled with a fluid of constant density. Initially, the system is at rest. All walls except the top (the "lid") are held fixed, so that the boundary conditions on these walls are $\boldsymbol{u} = (0, 0)$. At time $t = 0$, the lid is instantaneously prescribed the tangential velocity $\mathcal{U}$, so that the relevant boundary condition for $t > 0$ is $\boldsymbol{u} = (\mathcal{U}, 0)$. Without any other external forces ($\boldsymbol{f} = 0$), this drives a recirculating flow in the cavity, which reaches a steady-state solution for non-turbulent flow regimes. See figure 5.1 for an illustration of the experimental setup.

We consider the governing equations in dimensionless form by taking $\mathcal{L}$ and $\mathcal{U}$ as characteristic length and speed scales, and $\kappa(\mathcal{U}/\mathcal{L})^{n-1}$ as the characteristic viscosity. Since Newtonians, power law fluids and Bingham plastics are all special cases of the Herschel-Bulkley model, we use (2.32b), with the dimensionless Herschel-Bulkley

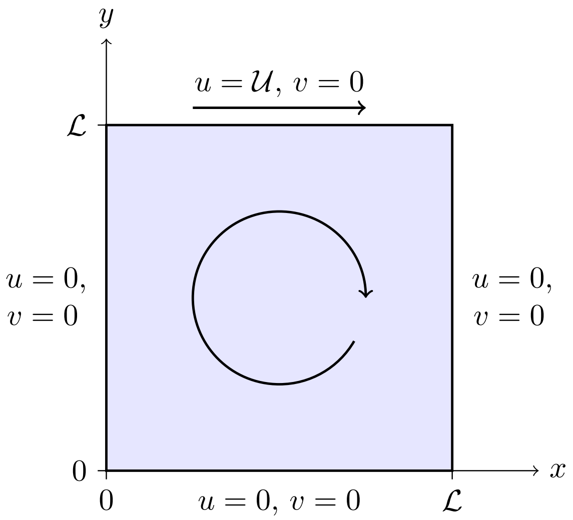


Figure 5.1: Schematic illustration of the lid-driven cavity test case.

apparent viscosity as given by (3.31). The relevant dimensionless groups are then

$$Re = \frac{\rho \mathcal{U}^2}{\kappa \left(\frac{\mathcal{U}}{\mathcal{L}}\right)^n}, \quad HB = \frac{\tau_0}{\kappa \left(\frac{\mathcal{U}}{\mathcal{L}}\right)^n}, \quad Pa = \frac{\mathcal{U}}{\varepsilon \mathcal{L}}. \tag{5.24}$$

Recall that the Reynolds number Re is the ratio of inertial forces to viscous ones, while the Herschel-Bulkley number HB quantifies the effect of yield stress versus power-law viscosity. It is worth noting that the Reynolds number for Newtonian flow ($n = 1$) is $Re_N = \rho \mathcal{U} \mathcal{L}/\mu$, since $\kappa = \mu$ in this case.

When we have information about the dimensionless apparent viscosity $\hat{\eta}$ throughout our domain at a given time, we can compute the local Reynolds number field $Re_L = Re/\hat{\eta}$. This value, which is proportional to the inverse of the apparent viscosity, provides insight about which regions of the domain are dominated by the effects of viscoplasticity, and which ones have near-Newtonian behaviour. Finally, for the special case $n = 1$, the Herschel-Bulkley number is replaced by the Bingham number $Bi = \tau_0 \mathcal{L}/\mu \mathcal{U}$, which allows comparisons with articles written on simulation of Bingham fluids.

In the remainder of this section, we use $Pa = 400$. It is important to choose a high enough value of Pa when using regularised schemes, and our choice is based on the arguments by Syrakos et al.[97], who show that this value results in highly accurate simulations for the lid-driven cavity problem. The cavity is discretised spatially with 256 cells in each dimension. The lid-driven cavity flow is then uniquely defined by the choices of Re, n and HB (or Bi, when $n = 1$).

Prior to our article[18], reference results for Herschel-Bulkley fluids had not been presented in the literature. There were, however, plenty of results for power-law fluids and Bingham plastics. Since these two characterise separate components of the Herschel-Bulkley model, we tested our code by validating each of them individually. Evaluation of the full Herschel-Bulkley model and presentation of our novel benchmark results are given in section 5.4.

Power-law fluids have zero yield stress, and are different from the Newtonian case when the flow behaviour index is different from unity. They were studied using a least-squares finite element formulation by Bell and Surana in 1994[157], who presented results with $n = {}^{1}\!/_{2}$ and $n = {}^{3}\!/_{2}$. A third-order finite volume method with upwinding was then used to study the same problem by Neofytou in 2005[158]. The classical way to compare results from the lid-driven cavity test case is to drive the system to steady state, and then look at the velocity profiles in each direction at slices perpendicular to the flow. This ensures that we can compare with results from codes that can only calculate steady-state solutions.

Figure 5.2 shows our results for power-law fluid velocity profiles, and includes reference results from the aforementioned papers. Additionally, we include the Newtonian case with comparisons from the classical paper by Ghia et al.[159]. It is evident that decreasing n results in less lid-induced kinetic energy propagating further down in the domain, whereas increasing it has the opposite effect. Our results align very well with those found in the literature, especially those due to Neofytou, which have the highest accuracy. We have also included the velocity profile for $n = 0.1$, in order to illustrate the significant viscous resistance in the limit $n \to 0$.

Due to the importance of accurately capturing viscoplastic behaviour, we have performed extra validation tests for the Bingham model. Figure 5.3 depicts velocity profiles at $Re = 100$ and various Bingham numbers, with comparison to the works of Neofytou[158] and Chai et al.[160]. In the latter reference, a multiple-relaxation-time lattice Boltzmann model was employed to simulate non-Newtonian fluids. We see that increasing the yield-stress has an effect on the velocity profiles similar to that of lowering n for the power-law fluids, although the Bingham fluids have sharper transitions in the profile (especially noteworthy in the x-velocity). Our results are in excellent agreement with the relevant references, and we have also extended the range of Bingham numbers beyond those for which velocity profiles are available in the literature.

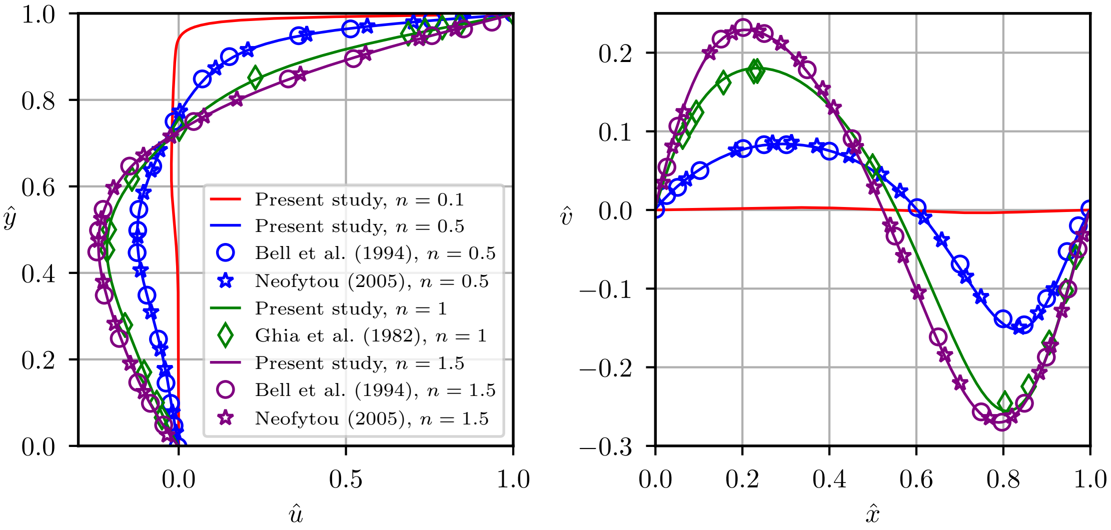


Figure 5.2: Steady-state velocity profiles for power-law fluids in the 2D lid-driven cavity, with $\kappa = 10\,\mathrm{Pa\,s}^n$. Left: First component of velocity through the vertical slice $\hat{x} = 0.5$. Right: Second component of velocity through the horizontal slice $\hat{y} = 0.5$.

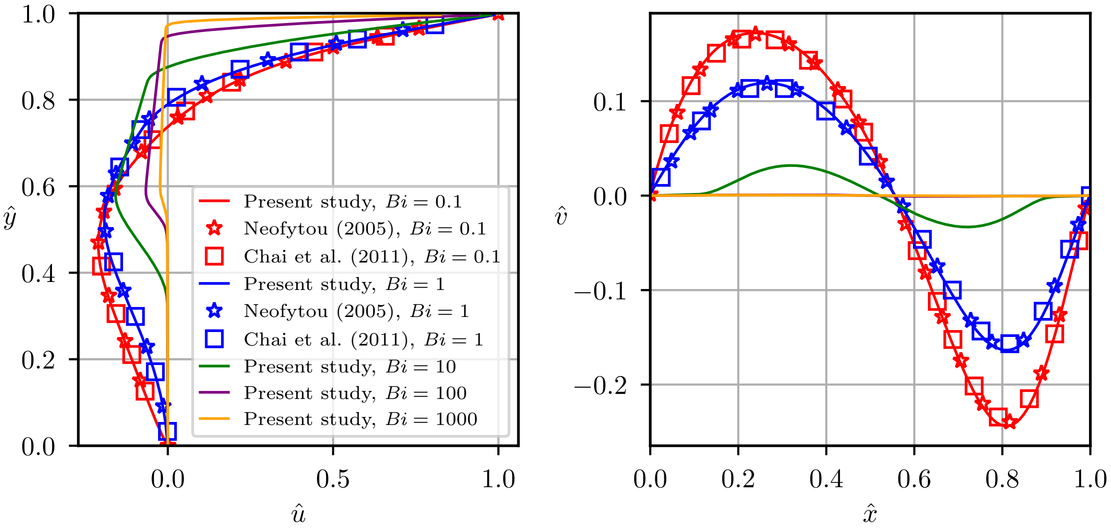


Figure 5.3: Steady-state velocity profiles for Bingham fluids in the 2D lid-driven cavity, with $Re = 100$. Left: First component of velocity through the vertical slice $\hat{x} = 0.5$. Right: Second component of velocity through the horizontal slice $\hat{y} = 0.5$.

The second method we use for validation is locating the position of the main vortex centre within the cavity, at steady-state. A comprehensive study exploring this dependency was performed in 2014 by Syrakos et al.[98]. We have performed simulations to steady-state for a large number of configurations of Re and Bi, and the resulting vortex locations are depicted in figure 5.4, alongside the results of Syrakos et al. We reiterate what they found in their paper: increasing the Bingham number moves the vortex upwards and to the right, while increasing the Reynolds number moves the vortex first towards the right, and then downwards and left to the centre. Our results agree very well with the reference results for the range of (Re, Bi) pairs covered in that study. Additionally, we have covered many more large Bingham numbers, obtaining results which follow the patterns to be expected. This illustrates the capability of our code to simulate fluids with very large yield stress. Please note that the number next to each point in the figure is the modified Reynolds number

$$Re^* = \frac{Re}{(Bi+1)}. \tag{5.25}$$

This is because, following Syrakos et al.[64], it provides a more natural measure of the transition to turbulence in the high-Bi region. For further discussions on choice of dimensionless parameters in viscoplastic fluid mechanics, we refer the reader to articles by de Souza Mendes[161] and Thompson and Soares[162].

The most important validation for viscoplastic fluids, is the accurate determination of yield surfaces. Finally, we would like to ensure that our time-marching scheme accurately captures the fluid movement. In order to validate both of these two attributes, we run a simulation starting from rest and evolving to the steady-state for the lid-driven cavity. Note that the criteria for the system having reached steady-state is that the maximum relative change in the stress field from one iteration to the next is less than some tolerance (in our case 10^{-5}). At this point, we set the lid velocity to zero, and let the system come to rest again through cessation. When the entire domain is an unyielded region, we stop the simulation. Although some authors have used time-stepping schemes to advance the system to steady-state[163;164], we are not aware of any results illustrating the yield surface development for instantaneous start from rest. On the other hand, the latter half of this numerical experiment, i.e. cessation of Bingham fluids from the lid-driven steady-state, was studied numerically by Syrakos et al. in 2016[64]. Comparisons with their yield surfaces and time measurements therefore serve as validation for our code.

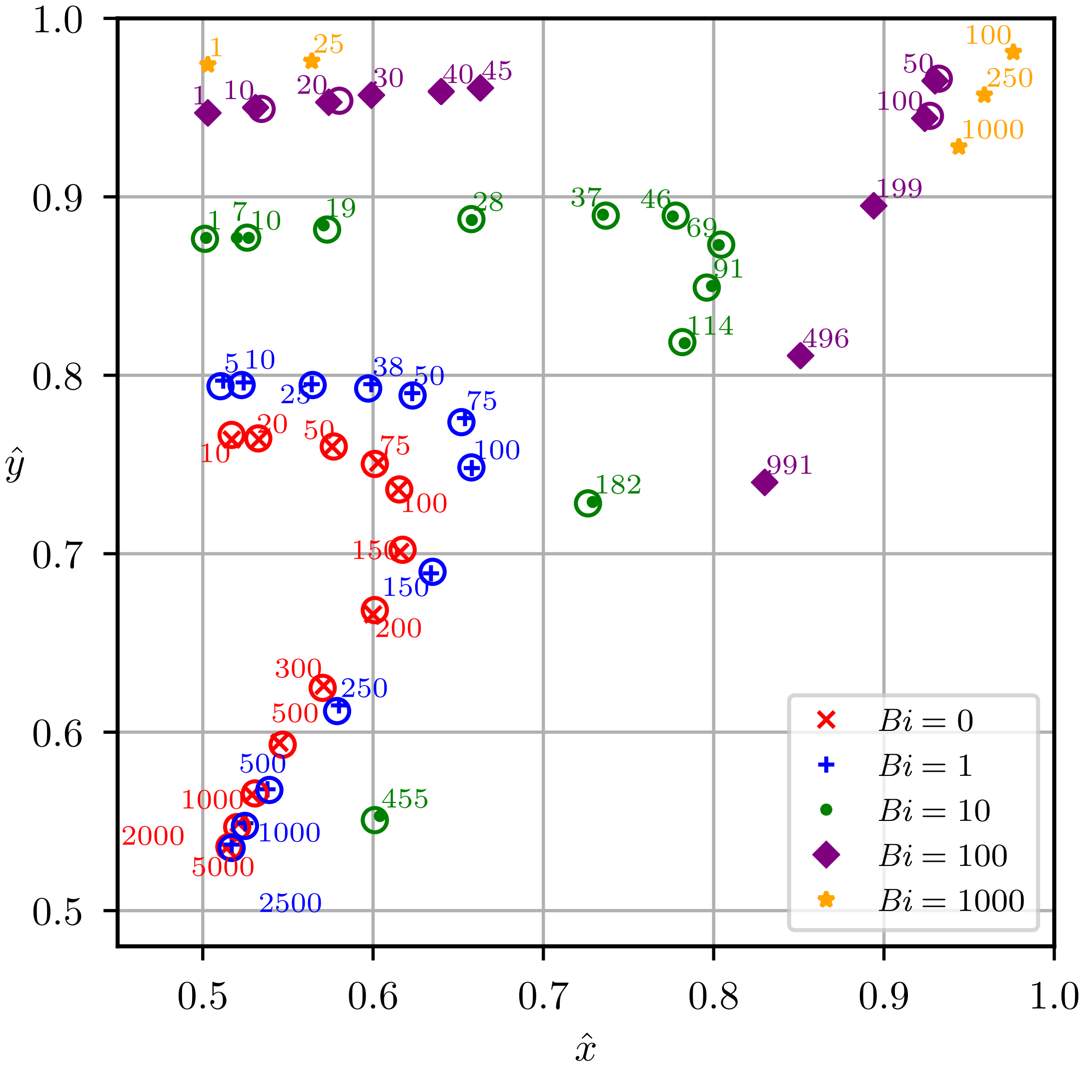


Figure 5.4: Steady-state vortex centre locations for Bingham fluids in the 2D lid-driven cavity. Empty circles are results from Syrakos et al.[98], while the remaining symbols are results from the present study. The number next to each point is the corresponding modified Reynolds number Re^*.

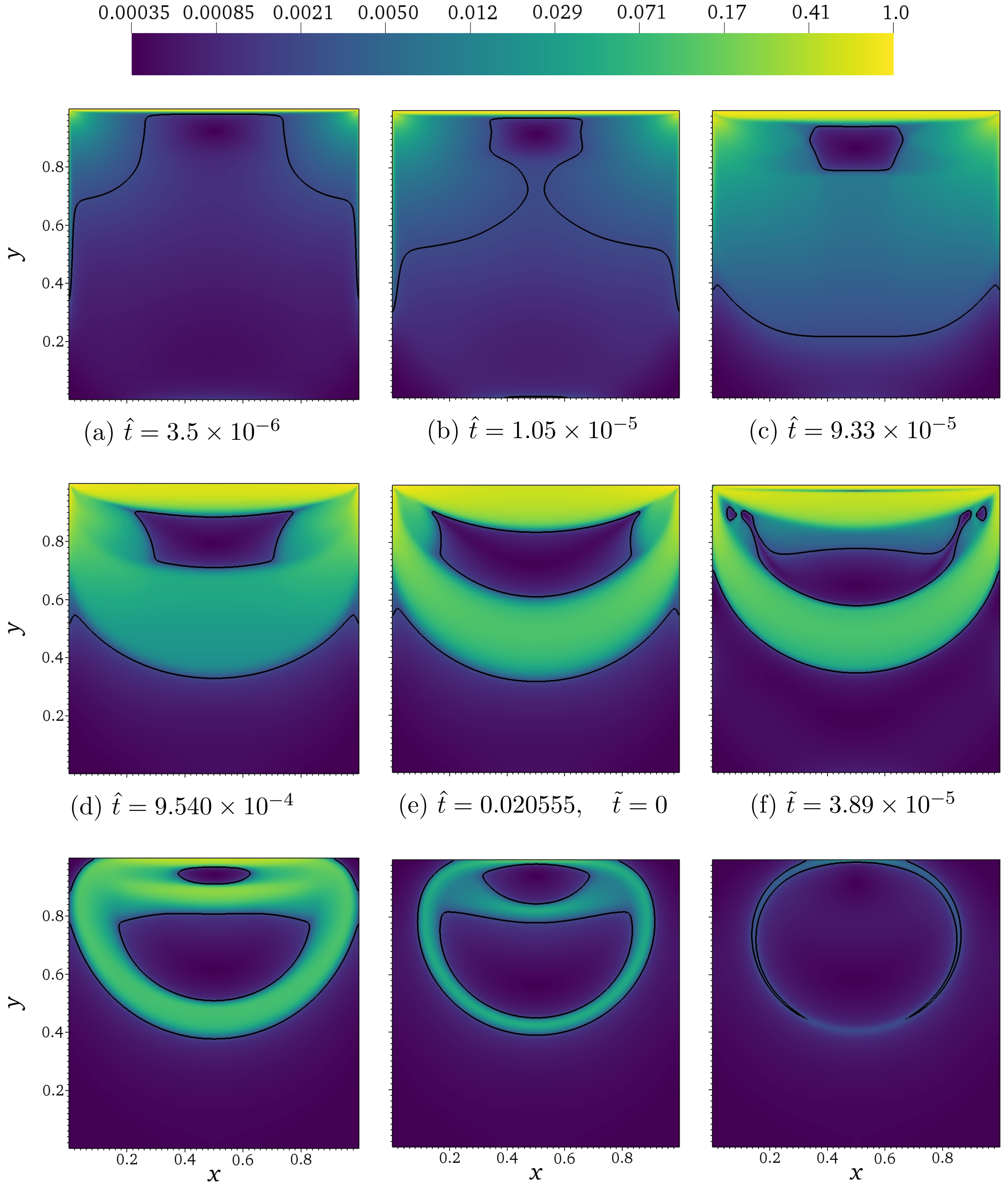
Results with $Re = 1$ and $Bi = 10$ are depicted in figure 5.5. The yield surface is illustrated as the single black contour line $\|\boldsymbol{\tau}\| = \tau_0$,[1] while the heat map shows the local Reynolds number Re_L. Note that the scaling of the color bar is logarithmic (see top of figure 5.5).

For viscoplastic fluids, unyielded regions occur in two distinct manners. Firstly, there are so-called stagnant zones, which are connected to no-slip walls and in which the fluid has zero velocity. On the other hand, unyielded zones with non-zero velocity are referred to as plug zones. These are not in contact with the walls, and rotate in the interior of the cavity. In fact, these two types of unyielded zones can never be in contact with each other, but must be separated by a layer of yielded fluid. This is because a plug zone connected to a stagnant zone would by association itself be stagnant, and thus could not move.

The first five plots (a)-(e) show the transition from rest to steady-state. It is clear that the fluid yields immediately at a very thin layer near the lid, since a large strain is applied here at $\hat{t} = 0$. Additionally, there are yielded regions which quickly propagate from the top corners and in towards the middle, eventually combining together and leaving an unyielded plug in the upper middle part of the cavity (a)-(c). This process occurs very quickly, in less than one percent of the time needed to reach steady-state, but should in fact happen instantaneously, following the discussion above. The initial connection between the stagnant zone at the bottom and the rotating plug is a spurious artefact due to the regularisation approach. Since we regularise the apparent viscosity, small but non-zero velocities are permissible in unyielded regions.

The steady-state shape of the resulting plug, and the unyielded region at the bottom of the cavity (e), is characteristic for the chosen pair of Reynolds and Bingham numbers. Comparisons to results by Syrakos et al. show that the steady-state yield surfaces are indistinguishable. At this point ($\hat{t} = 0.020552$), the lid is abruptly halted and held still. This removes the driving force for the flow from the system, and leads to cessation of the flow. Since cessation occurs in finite time for viscoplastic fluids, the cessation time t_c, defined as the time until the entire cavity is in the unyielded state, is an important characteristic. In order to measure $\hat{t}_c$, we introduce a new time $\tilde{t} = \hat{t} - 0.0205552$. We have chosen four times during flow cessation to compare with Syrakos et al.[64] and the flow patterns are very similar to their published results. Additionally, our simulations yield $\hat{t}_c = 1.2844 \times 10^{-3}$, which is a relative difference of less than 0.9%

[1] This simple method for determining the yield surface can be improved upon for regularised constitutive equations, as we discuss and illustrate in chapter 7.



(a) $\hat{t} = 3.5 \times 10^{-6}$

(b) $\hat{t} = 1.05 \times 10^{-5}$

(c) $\hat{t} = 9.33 \times 10^{-5}$

(d) $\hat{t} = 9.540 \times 10^{-4}$

(e) $\hat{t} = 0.020555, \quad \tilde{t} = 0$

(f) $\tilde{t} = 3.89 \times 10^{-5}$

(g) $\tilde{t} = 6.480 \times 10^{-4}$

(h) $\tilde{t} = 1.1664 \times 10^{-3}$

(i) $\tilde{t} = 1.2729 \times 10^{-3}$

Figure 5.5: Yield surface and local Reynolds number for time-dependent flow of Bingham fluids in the 2D lid-driven cavity. The fluid is initially at rest, and gradually transitions to a steady state (a)-(e). At this point, the lid is stopped, allowing cessation in finite time (e)-(i).

compared to the previously published results. Note also the unphysical connection between stagnant and plug zones in (i), which is in agreement with the results and discussions of Syrakos et al.[64]. In order to rectify these spurious artefacts, one would need a very high Papanastasiou number or use an unregularised approach.[165].

5.4 Herschel-Bulkley fluids

The results in section 5.3 validate our code for fluids obeying power-law and Bingham rheologies, but there are no available comparisons in the literature for Herschel-Bulkley fluids, which exhibit power-law dependencies in addition to a non-zero yield stress. Consequently, simulations were performed for viscoplastic fluids with the flow index n equal to 0.5 and 2.0. In order to cover a wide range of flow behaviours, we also chose to vary the Herschel-Bulkley number HB and the modified Reynolds number Re^*, as given by (5.25). For each combination of n, HB and Re^*, the system was advanced to steady-state, in order to obtain velocity profiles through the centre of the domain and the final location of the main vortex. These results can be used as benchmark references for codes simulating Herschel-Bulkley fluids.

Figure 5.6 shows the steady-state velocity profiles for each different parameter combination. As expected, a small flow index enhances the effect of the yield stress, resulting in weak velocity variations. This leads to the largely overlapping x-velocity profiles in the upper left plot, so that the y-velocity profiles in the upper right one are more useful comparisons. In the lower plots, however, where the fluid acts as a dilatant above the yield criterion, there is a trade-off between the effects of flow index and yield stress, so that we obtain large velocity variations throughout the middle of the domain in both directions.

In figure 5.7, we show the steady-state locations of the main vortex centres for each Herschel-Bulkley simulation. Comparing the results with figure 5.4, we see the same effects of increasing HB as for Bi, i.e. vortex moving upwards and to the right. Similarly, increasing Re^* results in the vortex moving first right and then down toward the cavity centre. Both of these observations are as expected, since the dimensionless groups capture similar aspects of the flow. Additionally, we note that decreasing the flow index n has a similar effect as increasing the yield stress, moving the vortex centre upwards and to the right. These additional data points serve as further benchmarking results for Herschel-Bulkley fluids.

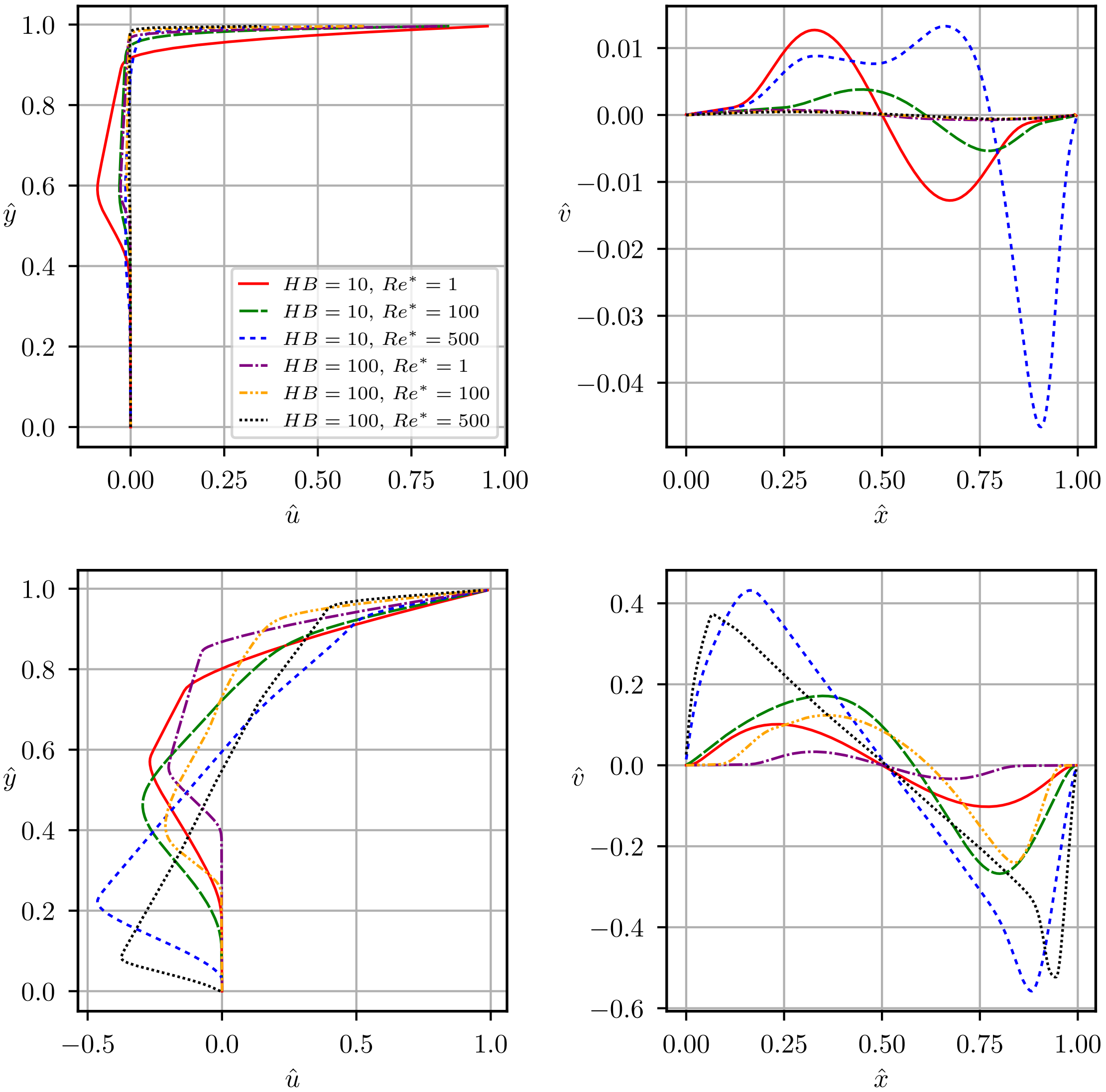


Figure 5.6: Steady-state velocity profiles for Herschel-Bulkley fluids with $n = 0.5$ (top) and $n = 2.0$ (bottom) in the 2D lid-driven cavity. The legend in the top left plot gives the combinations of *HB* and Re^*. Note that for the dilatants with $HB = 100$, several lines almost overlap.

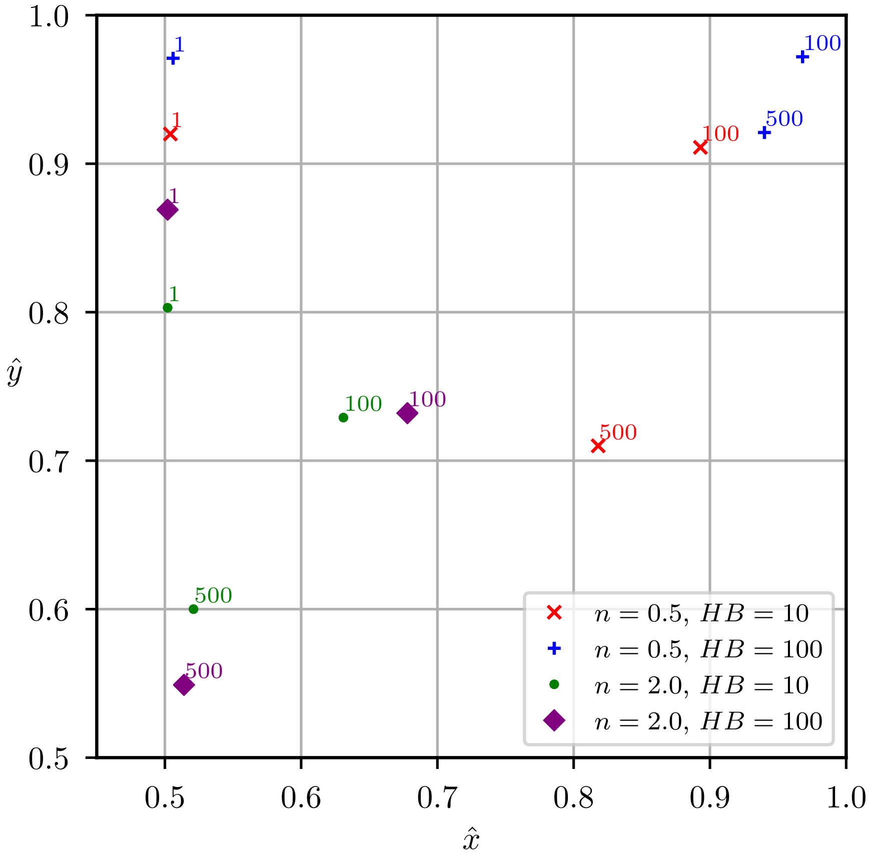


Figure 5.7: Steady-state vortex centre locations for Herschel-Bulkley fluids in the 2D lid-driven cavity with different combinations of flow index n, Herschel Bulkley number HB and modified Reynolds number Re^*. The number next to each point is the corresponding Re^*.

5.5 Applicability of adaptive mesh refinement

The use of AMR improves the efficiency of our algorithm in two different ways, depending on whether we are interested in steady-state results or transient behaviour. In the former case, we do not care about intermediate results, and would ideally employ a large time step. Unfortunately, due to the semi-implicit viscous solves employed in IAMR, we are unable to reach steady-state for highly resolved simulations by iterating with large time steps. The large system requires longer run times per time step, and the time step decreases proportionally with the grid size. On the other hand, we can rapidly advance a low-resolution simulation to steady-state, and then restart the simulation with one refinement level enabled throughout the entire domain. Due to the initial conditions, the finer resolution will converge after a much smaller amount of iterations. Subsequently, the simulation can be restarted again with a further level of refinement, and so on. In this manner, the computational expense associated with time-dependent simulation to a highly resolved steady-state is greatly reduced. As an example, advancing the 2D case with $Re = 1$, $Bi = 10$ to steady-state on a 16-core desktop computer took 83 minutes and 13 seconds when using a resolution of 256^2. By first advancing a low-resolution system with 64^2 cells to steady-state (33 seconds), and then refining the entire domain with two levels, the runtime for obtaining the same steady-state result is reduced to just under 15 minutes. Due to the built-in checkpoint functionality in AMReX, restarting simulations with different runtime options in this manner is straightforward.

In order to illustrate the usefulness of AMR in unsteady simulations, we refer to figure 5.8, which shows a snapshot during the evolution of a Bingham fluid with $Re = 1000$ and $Bi = 1$ from rest to steady-state with AMR enabled. The base grid consists of 64 cells in each direction, and two levels of refinement are used to achieve high resolution near the yield surface. This is done by tagging cells where $\|\boldsymbol{\tau}\| \leq 1.05\tau_0$ for refinement. Note that due to the patch clustering algorithm described in section 4.2, the area covered by refined grids reduce quite a lot from the first to the second level, even though the same refinement criterion is used for cell tagging. Consequently, we only need to use large amounts of computational resources in some regions. Due to the block-structured AMR implementation, the grid distribution is highly parallel. Additionally, the temporal subcycling in time ensures that we only need to decrease the step size in refined cells – not globally. It is important to remember that the potential gains of using AMR are highly dependent on disparate length scales - the technique is ideal for cases where the necessary domain size is much larger than the extent of features which need to be accurately resolved. Its efficacy is evident in the case shown in figure 5.8, since the yielded region is a small part of the domain.

5.6 Moving to three dimensions

Since the augmented IAMR code is efficient and parallelises well, it is possible to run unsteady simulations of fully three-dimensional viscoplastic flows. As an extension of the lid-driven cavity for Bingham fluids in 2D, we consider a unit cube with stationary no-slip conditions on the floor and all four walls, while the lid (at $\hat{z} = 1$) drives the flow. The spatial domain is discretised using 256 cells in each directions, i.e. 16,777,216 cells in total. The three-dimensional simulations were run prior to our enabling of adaptive mesh refinement for viscoplastic flows, and consequently static grids were employed. As before, the system is initially at rest. When steady-state is reached, we stop the motion of the lid and allow cessation to take place. Figure 5.9 illustrates how the yield surface evolves through time. Note that it is still computed as the contour $\|\boldsymbol{\tau}\| = \tau_0$, but in 3D we can actually visualise it as a surface. The snapshot times in figure 5.9 are similar to those used in figure 5.5, but vary by small factors. Note also here the presence of numerical artefacts due to regularisation, demonstrated by connected stagnant and plug zones. Qualitatively, the resemblance with the 2D case is striking, as the flow clearly goes through the same steps to reach steady-state and cessation. Having said that, the third dimension allows a richer picture of the shape of the yield surface. Especially fascinating is the importance of the four vertical edges, which play a similar role as the 2D corners, and lead to the yield surface stretching out in four directions from the centre. We emphasise that this is the first time fully three-dimensional yield surfaces have been presented for the lid-driven cavity case. Previously, the only 3D results are slices of rigid zones presented by Olshanskii[155] (finite differences, 262,144 cells), and the color plot slices presented in the bi-viscosity simulations performed by Elias et al.[166], (819,468 elements).

In figure 5.10, steady-state yield surfaces are shown for various combinations of Re and Bi. Readers who are familiar with the two-dimensional test case (see e.g. Syrakos et al.[98]) will recognise the similarity between those results and slices through these three-dimensional regions. Note that the slices are taken at a slight angle in the xy-plane. These simulations illustrate the code's ability to solve the governing PDEs

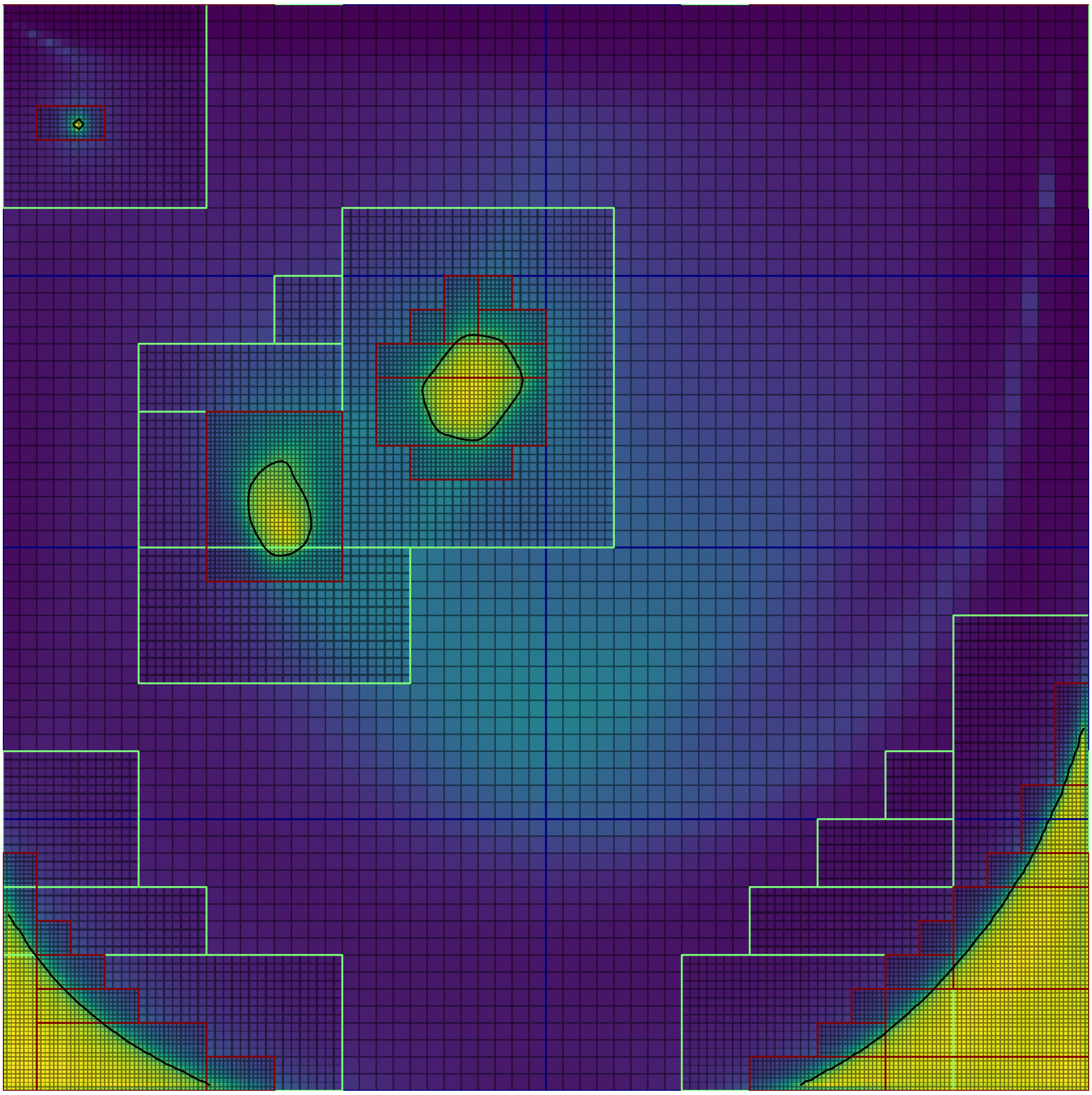
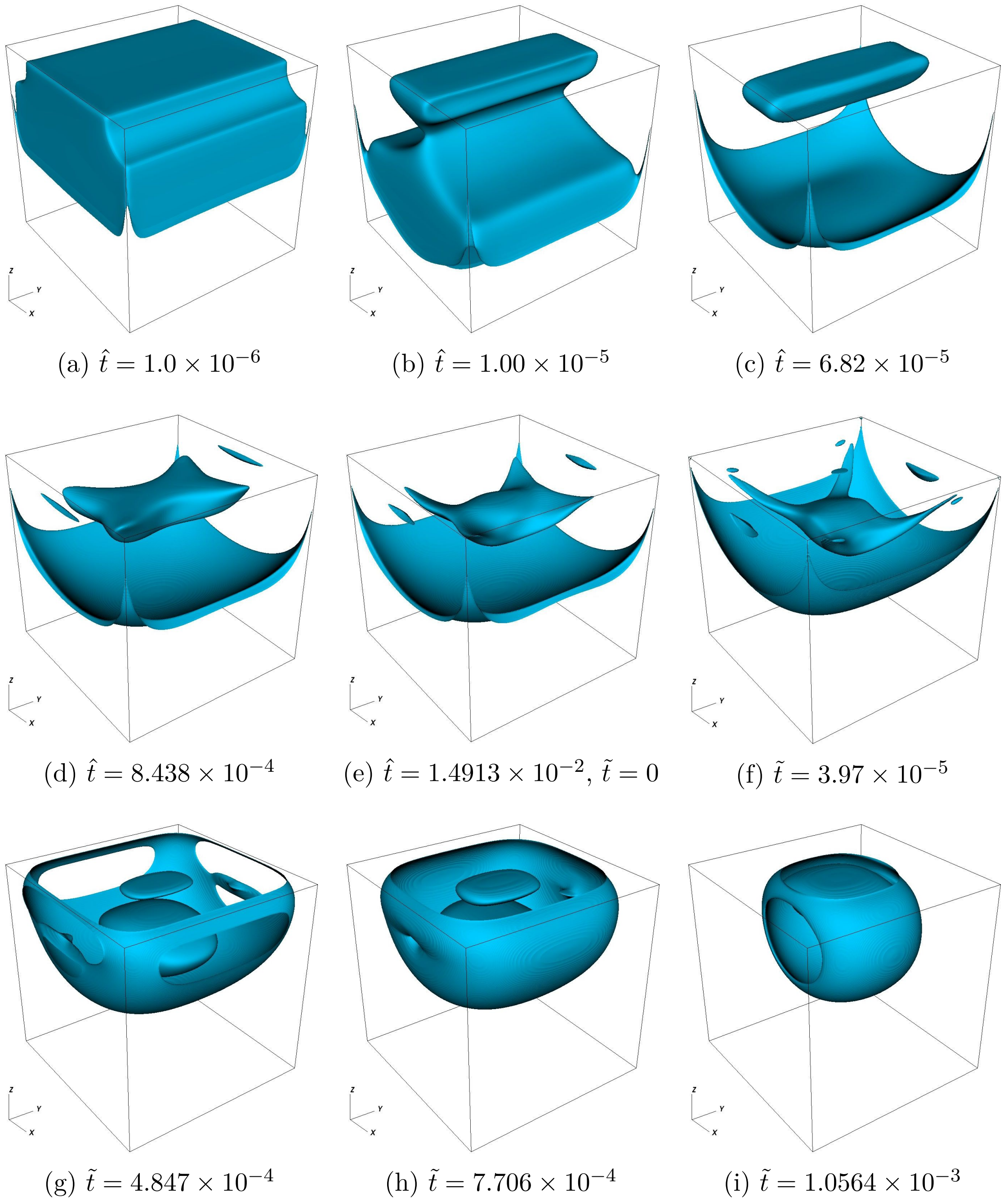


Figure 5.8: Illustration of spatio-temporal adaptive mesh refinement for tracking the development of the yield surface of a Bingham fluid with $Re = 1000$ and $Bi = 1$ in the 2D lid-driven cavity. The heatmap shows the magnitude of the stress tensor, which is what the grid refinement criterion is based on. The coarsest grid has cell spacing 1/64, but the two refinement layers ensure that the effective resolution near the yield surface is 1/256.



(a) $\hat{t} = 1.0 \times 10^{-6}$

(b) $\hat{t} = 1.00 \times 10^{-5}$

(c) $\hat{t} = 6.82 \times 10^{-5}$

(d) $\hat{t} = 8.438 \times 10^{-4}$

(e) $\hat{t} = 1.4913 \times 10^{-2}$, $\tilde{t} = 0$

(f) $\tilde{t} = 3.97 \times 10^{-5}$

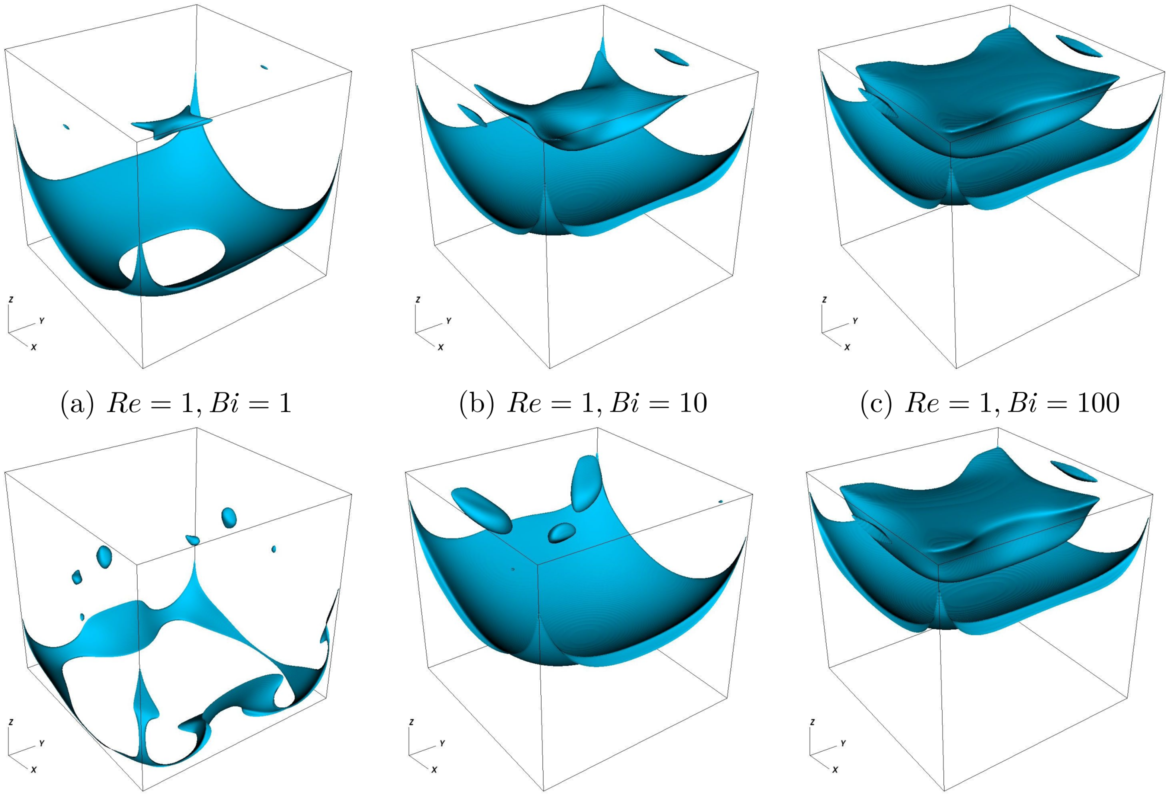
(g) $\tilde{t} = 4.847 \times 10^{-4}$

(h) $\tilde{t} = 7.706 \times 10^{-4}$

(i) $\tilde{t} = 1.0564 \times 10^{-3}$

Figure 5.9: Three-dimensional time-evolution of a Bingham fluid in the lid-driven cavity, with $Re = 1$ and $Bi = 10$. The system advances from an instantaneous start from rest to steady-state, before the lid motion ceases and the fluid stops. The lid moves in positive x-direction. The shapes of three-dimensional plug regions of this sort have never been identified for the lid-driven cavity before.

for time-dependent three-dimensional flows of non-Newtonian fluids following the Herschel-Bulkley model.



(a) $Re = 1, Bi = 1$ (b) $Re = 1, Bi = 10$ (c) $Re = 1, Bi = 100$

(d) $Re = 1000, Bi = 1$ (e) $Re = 1000, Bi = 10$ (f) $Re = 1000, Bi = 100$

Figure 5.10: Steady-state yield surface for Bingham fluids with various Reynolds and Bingham numbers in the 3D lid-driven cavity. The lid moves in positive x-direction.

While the shapes of the three-dimensional yield surfaces shown in figures 5.9 and 5.10 provide interesting insight about the three-dimensional effects, they do not provide quantitative comparisons for other codes for three-dimensional viscoplastic fluid flows. Consequently, we show the corresponding steady-state velocity profiles in figure 5.11. The first velocity component is considered on the line $\hat{x} = 0.5$, $\hat{y} = 0.5$, while the x- and z-directions are swapped for the third component. As is evident, there are notable similarities between the two-dimensional simulations, as expected. These profiles can act as benchmark results for future 3D yield stress simulation codes.

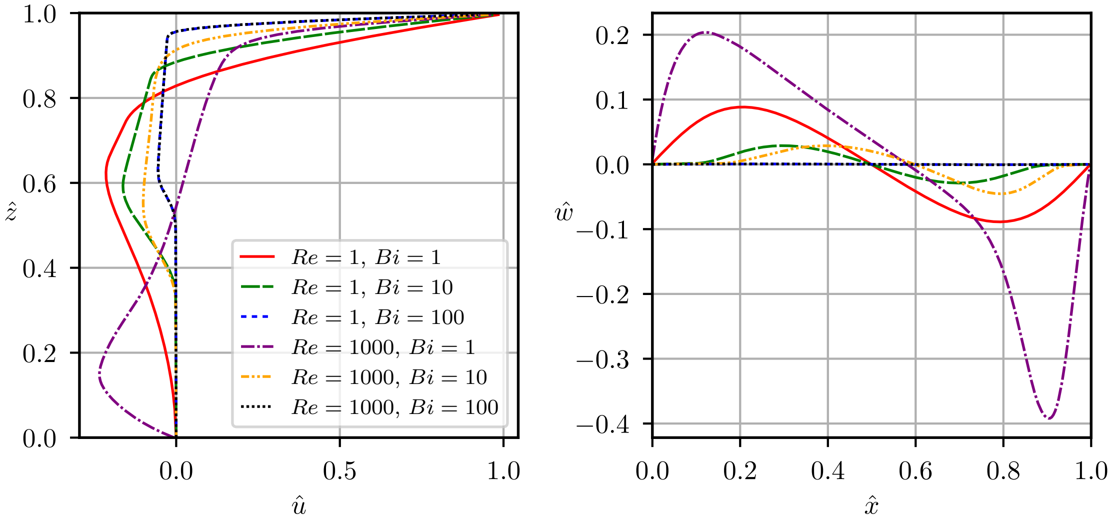


Figure 5.11: Steady-state velocity profiles for Bingham fluids in the 3D lid-driven cavity, provided as reference solutions for 3D viscoplastic solvers. The effect of Reynolds and Bingham numbers is clear. Note that the two cases with $Bi = 100$ are indistinguishable.

5.7 Parallel performance

We want to ensure that the excellent inherent scalability of IAMR is unaffected by the extension to generalised Newtonian fluids. As such, we run a test-suite with varying problem sizes and processor cores for the 3D lid-driven cavity test when the domain is filled by a Bingham fluid with $Re = 1$ and $Bi = 10$. In order to avoid the problems associated with strong and weak scaling, we vary the number of cores and problem resolution, but only use few cores for smaller problems and many cores for larger ones. Subsequently, we compute the rate for the cell updates, which gives proper insight into the resulting computational efficiency. In order to evaluate the parallel performance rather than the effect of system state, we use a constant time step, advect the system by 100 time steps, and measure the average runtime per time step for the next ten.

Figure 5.12 illustrates the results of average run times and rate for the scaling test. For a given problem size, the reduction in runtime (and increase in rate) is basically optimal until a core limit is reached. At this point, the MPI communication overhead becomes too large for further parallelisation, since the grid has already been broken down too far. However, as the problem gets bigger this core limit naturally increases, so that the computational rate can continue increasing as long as larger problems are solved. When simulating at a given problem resolution, it is therefore best to choose the lowest amount of cores which maximise the rate. Note that these tests do not utilise OpenMP tiling and have not been optimised in terms of runtime parameters for parallelisation (such as maximum size for subgrids), and still the scalability is excellent.

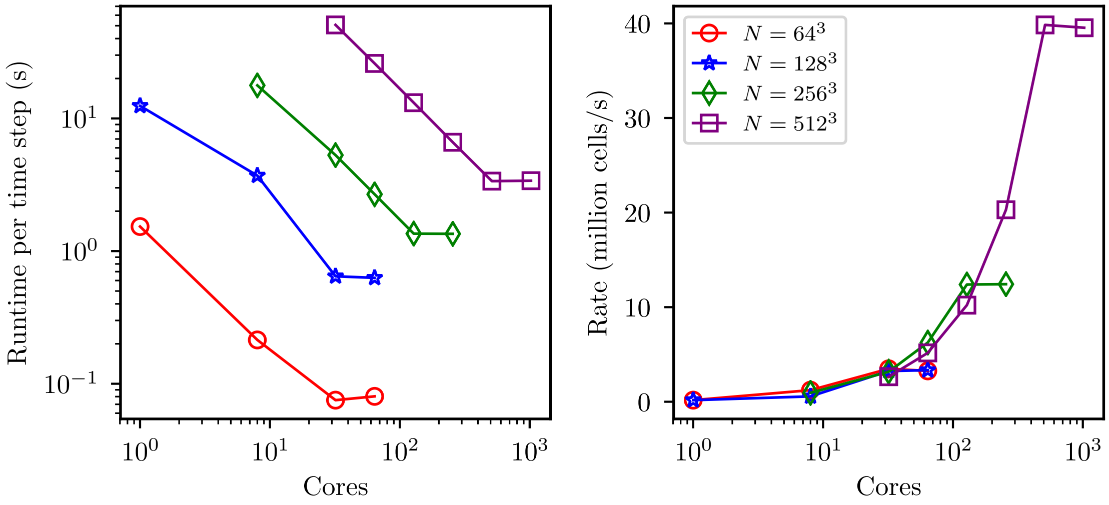


Figure 5.12: Strong scaling results for simulation of Bingham flow in the 3D lid-driven cavity. The left plot shows the reduction in runtime per time step as the amount of processor cores is increased, for various problem sizes. The right plot shows how the corresponding computation rate grows. Note that both axes are logarithmic in the left plot, while only the primary axis employs logarithmic scaling in the right one.

Chapter 6

Immiscible displacement flow

Fluid dynamics systems involving more than one fluid are countless, and many are so heavily characterised by the interactions that their inclusion in the system description is essential in order to understand it. For this reason, mathematical modeling and numerical simulation of several fluids has been actively researched for the last half century. There are numerous mathematical descriptions which capture multi-material flow, and for each of these a range of algorithms to solve the governing equations.

From a continuum fluid mechanics point of view, multi-material interactions are used to describe problems where there is a relatively sharp interface between two materials. Examples of such cases are water/oil separation (liquid-liquid interaction), penguins swimming through water (solid-liquid interaction) and water splashing into the air as a result of a surface disruption (liquid-gas interaction). In reality, all these interfaces are continuous from one material to the other. Moreover, because the length scale of the interface is far smaller than even the discrete control volumes used for simulation, it is a good approximation to treat the interfaces as relatively sharp.

The location of the boundary must be tracked as time evolves. The most common ways of doing this is by the use of advected indicators in the fluid, or using level set methods, as proposed by Osher and Sethian[167]. The level set function for a region is initiated such that it is negative inside the region, positive outside it and zero on the boundary. By taking into consideration the convection of the fluids under consideration, the level set function is then evolved so that its zeros move with the domain boundary. Osher and Fedkiw[168] provide an excellent overview of level set methods and their applications. We will not be delving further into the use such methods now, but will return to them when treating solid boundaries through the use of cut cells in chapter 7.

Secondly, any interaction between different fluids across the material interfaces, and chemical reactions occurring due to these interactions, must be modelled accurately. In order to do capture thermodynamically consistency in interactions, Fedkiw et al. developed the ghost fluid method[169]. In ghost fluid methods, each material has ghost cells on the side of the boundary where the other material exists. The method has since matured considerably[170–172]. Reactive flows are of special concern in scenarios of high temperature and pressure, since it can lead to phenomena such as combustion[173;174]. In our work, we are interested in non-reactive, highly immiscible interactions between fluids with different densities and rheological parameters. As such, we can limit ourselves to relatively simple methods in which a passive indicator carries the information about which fluid is present at each point in space and time.

Noh and Woodward[175] first introduced the advection of a scalar function used to indicate the volume fraction of each fluid in a cell, a method which was named the Volume Of Fluid (VOF) method by Hirt and Nichols[176] in 1981. Further development of the method by Sussman et al. allowed coupling to level-set methods[147], adaptive mesh hierarchies[177] and three-dimensional flow[178]. Ménard, Tanguy and Berlemont combined VOF with the ghost fluid method for gas interactions[179]. As such, the methodology is mature, versatile and well-documented, and is implemented in major commercial software suites for CFD. Since the interplay between displacing and displaced fluids is of the utmost importance in cement displacement simulations, we therefore needed to implement the methodology in our code.

6.1 Variable density: Rayleigh-Taylor instability

Before endeavouring to simulate displacement of a fluid with certain rheological properties by another one with different ones, it is worth pointing out that the projection method as implemented in IAMR allows variable density flow. In practice, this means that we can prescribe a density jump over a sharp interface in our domain, and thus simulate the effects of body forces acting on fluids with different densities. The Rayleigh-Taylor instability is a famous class of interface problems in which fingering occurs between two liquids of different density[180]. In its most simple form, it can consist of a heavy fluid located above a lighter one, with gravity acting on them both. Any slight perturbation in the interface between the fluids will lead to local pressure variations, which quickly develop into columns of displaced fluids.

In order to investigate this effect within the existing capabilities of IAMR, we simulate a test problem initially proposed by Tryggvason[181]. A two-dimensional domain $\Omega = [0, \mathcal{L}] \times [0, \mathcal{H}]$ is filled with a heavy fluid of density ρ_H on top of a lighter one of density ρ_L, with a small initial perturbation in the interface separating them. The initial perturbation of the interface is given by

$$y_p(x) = \frac{\mathcal{H}}{2} + \frac{\mathcal{L}}{10} \cos\left(\frac{2\pi x}{\mathcal{L}}\right). \tag{6.1}$$

With the gravitational acceleration g as the driving force, a characteristic velocity is $\mathcal{U} = \sqrt{g\mathcal{L}}$. Aside from the aspect ration $\mathcal{H}/\mathcal{L}$, there are two dimensionless groups of interest. Firstly, taking ρ_L as the characteristic density, the Reynolds and Galilei numbers reduce to the same quantity, since

$$Re = \frac{\rho_L \mathcal{U} \mathcal{L}}{\mu} = \frac{\rho_L g^{1/2} \mathcal{L}^{3/2}}{\mu} = \frac{\rho_L g \mathcal{L}^2}{\mu \sqrt{g\mathcal{L}}} = \frac{|\boldsymbol{f}| \mathcal{L}^2}{\mu \mathcal{U}} = Ga. \tag{6.2}$$

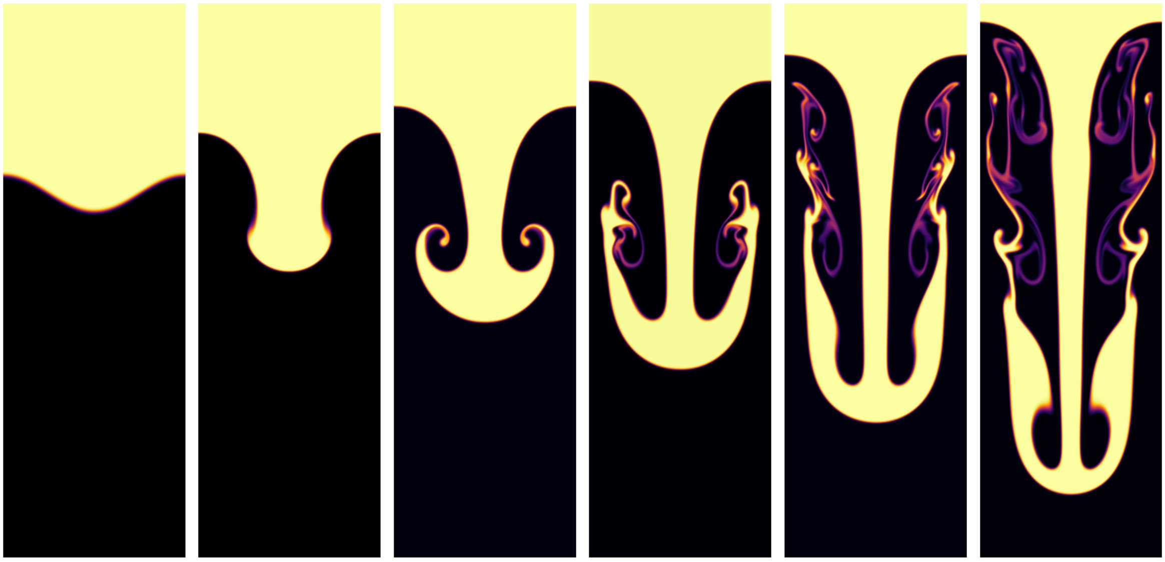
Additionally, the difference in densities is quantified through the Atwood number

$$\mathcal{A} = \frac{\rho_H - \rho_L}{\rho_H + \rho_L}. \tag{6.3}$$

Following Tryggvason[181] and later numerical studies by Guermond and Quartapelle[182] and Hosseini et al.[183], we take $\mathcal{H} = 4\mathcal{L}$ and $\mathcal{A} = 1/2$. Tryggvason considered the inviscid case, but since we have capabilities for solving viscous flows, we copy the latter two references and let $Re = 3000$. For temporal comparisons, we scale by Tryggvason's characteristic time length $\sqrt{g\mathcal{A}/\mathcal{L}}$, which is independent of viscosity. To avoid a sharp discontinuity in the initial conditions, the density at time zero is slightly smoothed out according to

$$\rho(x) = \rho_L + \frac{1}{2}(\rho_H - \rho_L)\left(1 + \tanh\left(\frac{y - y_p(x)}{\mathcal{L}/100}\right)\right). \tag{6.4}$$

Figure 6.1 shows how the initial perturbation evolves into a turbulent Rayleigh-Taylor instability. Comparisons of our results with those of Hosseini et al.[183], which are the most recent and accurate of the numerical studies in the literature, clearly show the same behaviour. However, our code has been run with a base grid of 128 cells in addition to two levels of adaptive mesh refinement on the interface, leading to a very high local refinement ($\Delta x = \mathcal{D}/512$). This allows us to capture much more detail in the regions which display turbulence.



(a) $\hat{t} = 0$ (b) $\hat{t} = 1.1156$ (c) $\hat{t} = 1.6641$ (d) $\hat{t} = 2.2151$ (e) $\hat{t} = 2.7686$ (f) $\hat{t} = 3.3215$

Figure 6.1: Temporal evolution of the density field in the Rayleigh-Taylor problem. The small initial perturbation develops into complex fingering structures displaying turbulent mixing of the heavy and light fluids.

In order to validate the results quantitatively, we have traced the topmost position of the light fluid and the bottom of the heavier one, as time evolves. Figure 6.2 displays our results alongside those of the other authors[181–183]. We match the results by Hosseini et al. perfectly. As described in their paper, the inviscid study by Tryggvason gave a slightly slower penetration of the heavier fluid downwards, while Guermond and Quartapelle saw a somewhat slower rise in the lighter fluid. We take the fact that our results match those produced in Hosseini et al., who utilised a refined algorithm for the Navier-Stokes-Cahn-Hilliard equations, as a strong validation for IAMR.

6.2 Volume of fluid method in IAMR

Presently, we further extend the capabilities of the IAMR code, to systems containing multiple generalised Newtonian fluids. This necessitates the use of a passively advected scalar to indicate the locations of each fluid as the system evolves in time. To ensure that the two-fluid extension works as expected, we validate our code against the two-dimensional displacement flow of a Bingham fluid by a Newtonian fluid along a plane channel. When the yield stress of the Bingham fluid is high enough, this generates static wall layers, as illustrated by Wielage-Burchard and Frigaard[184].

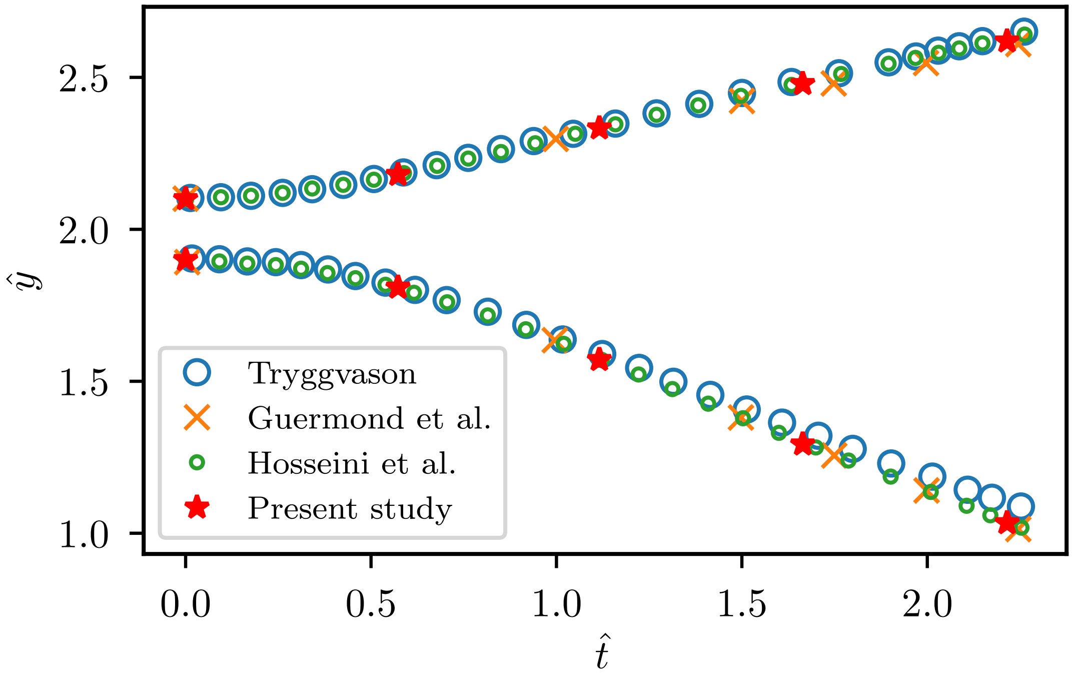


Figure 6.2: Comparisons of our results with those in the literature for front tracking of each of the two fluids. Tryggvason utilised Lagrangian-Eulerian vortex methods for inviscid flow in 1988[181]. Guermond and Quartapelle then implemented a second-order finite element projection algorithm in 2000[182], while Hosseini et al. applied Galerkin-based isogeometric analysis to the Navier-Stokes-Cahn-Hilliard equations in 2017.

In order to describe immiscible two-fluid flow, we introduce a passive scalar $\lambda(\boldsymbol{x}, t) \in [0, 1]$ which plays the role of an indicator, and let the simulated fluid properties be a function of this indicator. In addition to extending (2.29) by an advection equation for λ, the apparent viscosity must now depend on this scalar. For this system, the fluid motion is therefore governed by

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \cdot \nabla \rho = 0, \tag{6.5a}$$

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}, \tag{6.5b}$$

$$\frac{\partial \lambda}{\partial t} + \boldsymbol{u} \cdot \nabla \lambda = 0, \tag{6.5c}$$

$$\boldsymbol{\tau} = \eta \left(\|\dot{\boldsymbol{\gamma}}\|, \lambda \right) \dot{\boldsymbol{\gamma}}, \tag{6.5d}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{6.5e}$$

completed by a suitable constitutive equation as introduced in section 2.2. In general, (6.5c) includes a diffusive term proportional to $\nabla^2 \lambda$ on the right hand side, in addition to a forcing term, but for the case of immiscible and non-reactive flow which we are studying, they are both zero.

Finally, we need to ensure that the model encapsulates the dependency of the apparent viscosity on which fluid is present at a given point. The extension of viscoplastic flow in the IAMR framework to incorporate several fluids is the main motivation for the present work. For Newtonian fluids, whose viscosity is given by a constant dynamic coefficient μ_i in the fluid i, this is done by linear interpolation based on λ. For a system of two fluids, $i \in \{1, 2\}$, we therefore take

$$\mu(\lambda) = \lambda \mu_1 + (1 - \lambda) \mu_2, \tag{6.6a}$$

so that $\mu = \mu_1$ when $\lambda = 1$ and $\mu = \mu_2$ when $\lambda = 0$. For Herschel-Bulkley fluids, which use two additional parameters to describe the apparent viscosity, we simply take similar averages to obtain the apparent flow index and yield stress:

$$n(\lambda) = \lambda n_1 + (1 - \lambda) n_2, \tag{6.6b}$$

$$\tau_0(\lambda) = \lambda \tau_{0,1} + (1 - \lambda) \tau_{0,2}. \tag{6.6c}$$

6.3 Plane channel displacement flow

As a validation problem, we consider the displacement of a Bingham fluid by a Newtonian in a plane channel, as described by Wielage-Burchard and Frigaard[184]. Our domain is a two-dimensional duct of width $2\mathcal{W}$ and length $\mathcal{L}$. Through the duct, there is a flow with mean x-velocity $\mathcal{U}$ across its width. In the absence of external forces, this corresponds to a plane Poiseuille flow, for which analytical solutions exist both for Newtonian and Bingham fluids. Initially, the domain is filled with a Bingham fluid, indicated by $\lambda = 1$. The properties of this fluid are $\mu_1 = 1\,\text{Pa\,s}$ and $\tau_{0,1} = 100$ Pa. We initialise the velocity profile as the steady-state analytical solution for Bingham fluids, given by

$$u(\hat{y}) = u_{\max,1} \cdot \begin{cases} 1, & |\hat{y}| \leq y_0, \\ 1 - \left(\frac{|\hat{y}|-y_0}{1-y_0}\right)^2, & y_0 < |\hat{y}| \leq 1, \end{cases} \tag{6.7}$$

where

$$u_{\max,1} = \frac{(1-y_0)^2}{2y_0} \cdot \frac{\tau_{0,1}\mathcal{W}}{\mu_1} \tag{6.8}$$

is the maximum velocity of the fluid, occurring in the plug region $|\hat{y}| < y_0$. Note that we have scaled lengths by the channel half-width $\mathcal{W}$. The characteristic dimensionless distance y_0 is found by demanding $\int_0^1 u(\hat{y})\mathrm{d}\hat{y} = \mathcal{U}$, resulting in the cubic equation

$$y_0^3 - 3\left(1 + \frac{2\mu_1\mathcal{U}}{\tau_{0,1}\mathcal{W}}\right) y_0 + 2 = 0, \tag{6.9}$$

which can be solved numerically for y_0.

At time $t = 0$, we inject a Newtonian fluid with dynamic viscosity $\mu_2 = 10$ Pa s at the left-hand side of the duct. This is achieved through an inflow boundary condition where $\lambda = 0$ and the velocity profile is

$$u(\hat{y}) = \frac{3}{2}\mathcal{U}(1-\hat{y})^2). \tag{6.10}$$

We set $\mathcal{W} = 1$ m, $\mathcal{L} = 10$ m, $\mathcal{U} = 1$ m/s, $Pa = 200$ and use 128 cells across the channel width to obtain high-resolution color plots. Figure 6.3 shows the resulting distributions of the indicator λ, x-velocity magnitude u and strain rate magnitude $\|\dot{\boldsymbol{\gamma}}\|$ after 3.4 s. Note that due to the high Bingham number ($Bi = 100$), residual wall layers of fluid 2 remain along the top and bottom walls. The velocity profile follows the Bingham profile downstream of the displacement front (for this case, we have

$y_0 \approx 0.86$), while a quadratic profile develops between the stationary residual wall layers upstream of the front.

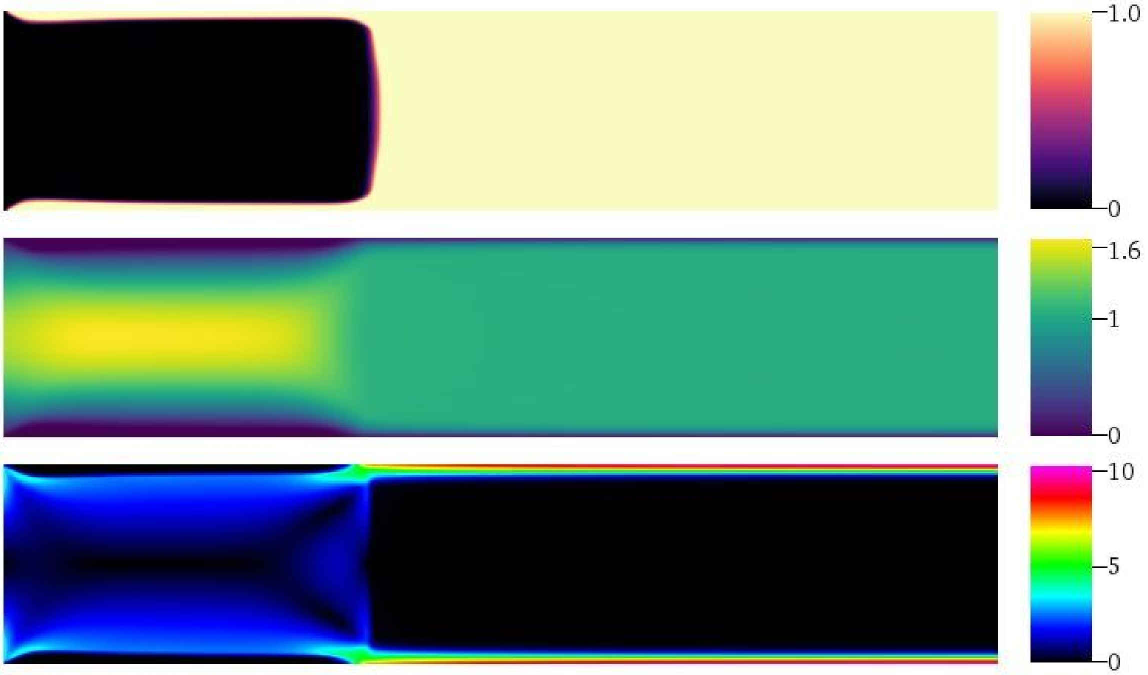


Figure 6.3: Snapshots of the plane channel displacement flow at $t = 3.40\,\mathrm{s}$, depicting the indicator (top, dimensionless), x-velocity (middle, m/s) and strain-rate magnitude (bottom, s^{-1}). In each case, the distribution in the full duct $[0, \mathcal{L}] \times [-\mathcal{W}, \mathcal{W}]$ is shown.

In order to obtain a quantitative comparison with the results of Wielage-Burchard and Frigaard[184], we run the simulation until steady-state with 32 cells across the channel width. Slices through the centre of the domain are shown in figure 6.4, with the reference results superimposed. As is evident, there is excellent agreement between our model and the reference. The residual wall layer is clearly captured, inside of which the parabolic profile shows the coexistence of Newtonian and Bingham fluids in the channel.

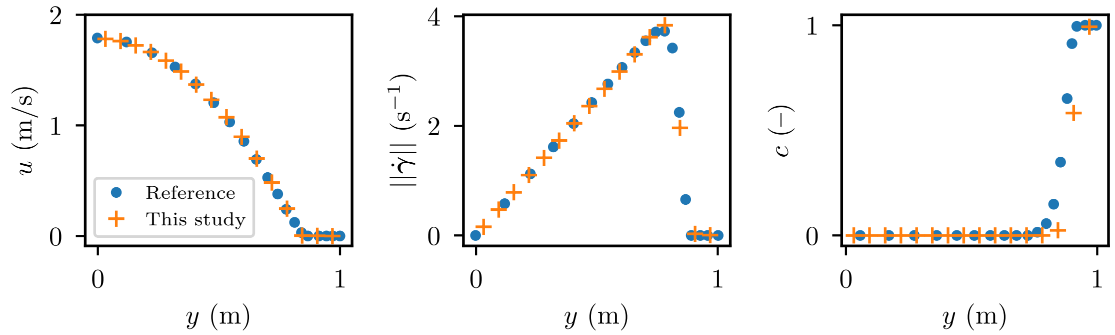


Figure 6.4: Comparisons of plane channel displacement results with Wielage-Burchard and Frigaard[184] for validation of the volume of fluid implementation in our our code. We look at the variation at $x = 5$ m. Our indicator function is sharper than the reference, but we obtain the same static layer width, as is evident from the profiles of x-velocity and strain-rate magnitude.

Chapter 7

Embedded boundaries for complex domain geometries

Perhaps the most important feature apart from the ability so solve the governing equations, is the ability to do so in and around geometrically interesting shapes. Although the work in previous chapters demonstrated how state-of-the-art computational techniques facilitate simulations which were previously considered too expensive, the code did not have any way of dealing with non-trivial domain geometries, and consequently could only be used to simulate flow in simple domains (rectangles in two dimensions and right rectangular prisms in three). We therefore extend the software package to non-trivial geometries through the use of embedded boundaries (EBs). This was a natural extension of our work, and the embedded boundary approach was particularly well-suited to the structured mesh discretisation. Additionally, the AMR capabilities of the underlying software suite, AMReX, go hand in hand with embedded boundaries, allowing viscous boundary layers to be fully resolved through localised mesh refinement.

The need for new and improved numerical tools for simulating viscoplastic flow phenomena is especially clear for problems with geometrical structures playing an important role, such as the recent work by Koblitz et al.[185], which would not be practically possible within current software packages without considering the 2D steady-state problem. Having said that, there are notable advances being made, such as a study on the transition to turbulence of viscoplastic fluids past a cylinder in 3D, which was simulated using Papanastasiou regularisation for small Bingham numbers[186]. A recent example of unsteady flow with fluid particle interactions is the study of time-dependent hydrodynamic interaction of 2D particles by Chaparian et al.[187]

Numerical simulations in the presence of complex boundaries can broadly be sorted by whether they use structured or unstructured grids. Both types of grids are widely utilised, and an overview of the usage of each in published simulations of yield stress fluids is provided in the review by Saramito and Wachs[89]. By definition, the difference between structured and unstructured grids is that the former requires regular connectivity of cells, while the latter does not. Consequently, the two types of grids have different advantages. Unstructured meshes have the advantage of greater geometric generality, while structured grids, which can incorporate geometric complexity with an embedded boundary approach, require storage of less geometric information. Algorithms on structured meshes can therefore exploit the excellent data storage for higher computational efficiency.

Non-Cartesian geometries are cut out of the underlying grid by storing local data representing the interface within each cell which contains part of the geometry. The methodology has several attractive qualities: rapid mesh generation regardless of complex geometries[188]; avoidance of locally skewed grids; inherent compatibility with quad- and octree adaptive mesh refinement (AMR); and retainment of the efficient and user-friendly data storage associated with structured grids. Special consideration must be taken to augment computational stencils and other algorithmic tools for and near the cut cells, but the technique nevertheless offers a rapid and relatively simple way of incorporating non-rectangular geometries without sacrificing the efficiency associated with structured grids.

Over the last decades, the embedded boundary method has matured and become a robust tool. The idea was first used by Purvis and Burkhalter in 1979[189], and subsequently, Wedan and South[190] to solve potential flow problems. Throughout the 1980s, the method was extended to solve the compressible Euler equations[24;191;192], which forms the basis for the general hyperbolic treatment of Colella et al. which we follow[193]. Since we are studying an incompressible system, we also require the solution of Poisson equations, which was published by Johansen and Colella in 1998[194]; a feat which they and McCorquodale naturally extended to solutions of the heat equation a few years later[195]. Developing novel and improved schemes for embedded boundaries is still an active area of research[196–198].

Designed for computational speed and robustness, the AMReX framework facilitates efficient usage of Cartesian structured grids, and has extensive built-in support for EBs. We thus saw an opportunity to implement the first viscoplastic flow solver within such a framework, which one would expect to perform very well[153].

7.1 Revisiting the incompressible flow solver

When studying the background literature for embedded boundaries, it quickly became evident that although cell-wise operations only require alteration in cut cells and their neighbours, the required alterations within these cells grow rapidly with the complexity of the computations in regular cells. Specifically, algorithmic steps like the Godunov-type extrapolation and upwinding, which in any event are designed for convection-dominated flows, are unnecessarily complicated to implement for the problems which we are interested in. Additionally, the development of EB tools for AMReX in general were in its early stages when we started working on this, so incorporating all the functionality directly into a code as large and complex as IAMR was not a logical first step. For these reasons, we started the EB work in a new code called incflo, which, although similar to IAMR in its fundamental algorithm, has simpler procedures for convective terms and predictor/corrector updates, and did not have to inherit such a large amount of legacy code. A simple analysis of the relevant repositories gives a case in point: incflo consists of approximately 10,000 lines of code, whereas IAMR has 66,000. Moreover, we used all the previous verification and validation tests for IAMR during development of incflo, and found no loss of accuracy.

For the sake of clarity, we omit communication across refinement levels in the following description of the incompressible flow solver. Although the functionality required for multiple adaptive refinement levels is contained in the code we developed, the focus of our research was to enable viscoplastic flow simulation with an embedded boundary approach. This was ambitious enough without mixing in AMR usage, which would entail an extra level of debugging complexity and require additional validation. Consequently, we prioritised single-level simulations in the algorithmic development, and although we performed some simulations with refinement based on volume fraction, it was only for Newtonian flows and not extensive enough to include in our results. Instead, we focus on the steps required to move the system forward one time step on a single refinement level. Although the software is general enough to handle variable-density incompressible flows through a conservative advection of ρ, we only consider cases where it is constant, and consequently do not discuss conservative density updates, which in any event are trivial compared to the velocity terms.

We follow a MOL approach as discussed in section 3.1, in which the momentum equation as given by (2.29b) is only discretised in time, so that Runge-Kutta type schemes for ordinary differential equations (ODEs) can be utilised for the temporal advancement. We affix a superscript to the simulated variables at the m-th time step, so that the time itself is t^m and the velocity profile $\boldsymbol{u}^m$. Algorithm 7.1 gives the pseudocode to advance the simulation from time step m to $m+1$, as a reference for our upcoming discussion.

Algorithm 7.1 Advance simulation one time step

```
function Advance($t^m, \boldsymbol{u}^m, p^m$)
    ApplyBoundaryConditions($\partial\Omega$)
    $\Delta t :=$ TimeStepSize($t^m, \boldsymbol{u}^m, p^m$)
    procedure Predictor
        $\boldsymbol{C}^m, \eta^m, \boldsymbol{V}^m :=$ DerivedQuantities($\boldsymbol{u}^m$)
        $\boldsymbol{C} \leftarrow \boldsymbol{C}^m$
        $\eta \leftarrow \eta^m$
        $\boldsymbol{V} \leftarrow \boldsymbol{V}^m$
        $\boldsymbol{\psi} \leftarrow \rho\boldsymbol{u}^m + \Delta t\,(\rho\boldsymbol{C} + \boldsymbol{V} - \nabla p^m + \boldsymbol{f})$
        $\tilde{\boldsymbol{u}}^* :=$ UpdateVelocity($\Delta t, \eta, \boldsymbol{\psi}$)
        $\boldsymbol{u}^*, p^* :=$ NodalProjection($\Delta t, \tilde{\boldsymbol{u}}^*$)
    end procedure
    procedure Corrector
        $\boldsymbol{C}^*, \eta^*, \boldsymbol{V}^* :=$ DerivedQuantities($\boldsymbol{u}^*$)
        $\boldsymbol{C} \leftarrow \frac{1}{2}\,(\boldsymbol{C}^m + \boldsymbol{C}^*)$
        $\eta \leftarrow \frac{1}{2}\,(\eta^m + \eta^*)$
        $\boldsymbol{V} \leftarrow \frac{1}{2}\,(\boldsymbol{V}^m + \boldsymbol{V}^*)$
        $\boldsymbol{\psi} \leftarrow \rho\boldsymbol{u}^m + \Delta t\,(\rho\boldsymbol{C} + \boldsymbol{V} - \nabla p^* + \boldsymbol{f})$
        $\tilde{\boldsymbol{u}}^{m+1} :=$ UpdateVelocity($\Delta t, \eta, \boldsymbol{\psi}$)
        $\boldsymbol{u}^{m+1}, p^{m+1} :=$ NodalProjection($\Delta t, \tilde{\boldsymbol{u}}^{m+1}$)
    end procedure
    $t^{m+1} := t^m + \Delta t$
    $m \leftarrow m + 1$
end function
```

After applying boundary conditions and computing the time step size according to (4.17), with the viscous Courant number replaced by (5.13), as derived in chapter 5, we perform a two-step Runge-Kutta scheme with a predictor and a corrector step. The pseudocode in each of the procedures Predictor and Corrector has been written so as to highlight the similarities between them. Although we are dealing with a multi-step algorithm including projections within each procedure, this elucidates the classic Runge-Kutta scheme underlying the temporal integration.

The first step within each of the procedures is to compute the quantities which are directly derived from the velocity field and its gradients. The relevant pseudocode is given in algorithm 7.2. In order to capture convection within the fluid, a slope-based upwinding procedure is utilised. However, in contrast to the Taylor expansion procedure utilised to extrapolate variables to cell faces at $t^{m+1/2}$ in IAMR, we linearly interpolate the slopes directly, which is much easier to implement and computationally less intensive. Velocity slopes are computed within each cell by comparing values in neighbouring cells, according to the procedure outlined in 3.2. With one extrapolated value from each of the two cell adjacent to the face, a simple upwinding algorithm is utilised. The normal velocity components are treated first, and subsequently used as the background field when upwinding the transverse components.

These upwinded, extrapolated values at faces allow us to compute the convective term $\boldsymbol{C} = -\boldsymbol{u} \cdot \nabla \boldsymbol{u}^F$ as a flux term, since, by the chain rule, we have .

$$\nabla \cdot \left(\boldsymbol{u} \otimes \boldsymbol{u}^F \right) = \boldsymbol{u} \cdot \nabla \boldsymbol{u}^F + (\nabla \cdot \boldsymbol{u})\, \boldsymbol{u}^F = \boldsymbol{u} \cdot \nabla \boldsymbol{u}^F, \tag{7.1}$$

where the last equality holds due to the incompressibility constraint in (2.19). Before doing this, however, we must enforce that constraint by projecting the face-based extrapolated velocity $\tilde{\boldsymbol{u}}^F$ onto the space of solenoidal vector fields. This is necessary since the values extrapolated by upwinding the slopes are not necessarily divergence-free. Recalling section 3.3, the Helmholtz decomposition allows us to write any vector field as a sum of solenoidal and irrotational parts, so we define

$$\tilde{\boldsymbol{u}}^F = \boldsymbol{u}^F + \nabla \phi, \tag{7.2}$$

where $\boldsymbol{u}^F$ has zero divergence and $\nabla \phi$, being a scalar gradient, is irrotational. Taking the divergence of (7.2) leads to a Poisson equation for ϕ,

$$\nabla^2 \phi = \nabla \cdot \tilde{\boldsymbol{u}}^F, \tag{7.3}$$

which is straightforward to solve and which allows us to compute the divergence-free velocity field $\boldsymbol{u}^F$. Since the three components of $\boldsymbol{u}$ are extrapolated to separate faces, the unknown scalar ϕ must be cell-centred in order for its gradients to end up on the corresponding faces. For this reason, this projection is referred to as a cell-centred projection, although it has previously been referred to as the MAC projection due to its historical links to the spatial discretisation in the original marker and cell method[199].

Algorithm 7.2 Compute derived quantities: convective term, apparent viscosity and explicit viscous term

```
 1: function DerivedQuantities($\boldsymbol{u}$)
 2:     $s$ =ComputeSlopes($\boldsymbol{u}$)
 3:     $\tilde{\boldsymbol{u}}^F$ = ExtrapolateToFaces($\boldsymbol{u}, s$)
 4:     $\boldsymbol{u}^F$ = CellCentredProjection($\tilde{\boldsymbol{u}}^F$)
 5:     $\boldsymbol{C} = -\boldsymbol{u} \cdot \nabla \boldsymbol{u}^F$                          ▷ Convective term
 6:     $\|\dot{\boldsymbol{\gamma}}\| = \left\| \nabla \boldsymbol{u}^F + \left(\nabla \boldsymbol{u}^F\right)^\top \right\|$        ▷ Strain rate magnitude
 7:     $\eta = \eta(\|\dot{\boldsymbol{\gamma}}\|)$                             ▷ Apparent viscosity
 8:     $\boldsymbol{V} = \frac{1}{\rho}\nabla \cdot \left(\eta \left(\nabla \boldsymbol{u}\right)^\top\right)$          ▷ Non-linear stress divergence
 9:     return $\boldsymbol{C}, \eta, \boldsymbol{V}$
10: end function
```

With the convective terms in place, the next step is to compute the viscous terms. Since viscous terms dominate in many of the problems we are interested in, and since they can lead to overly restrictive time step criteria when treated explicitly, we opt for a semi-implicit temporal discretisation. Rather than sticking to the method employed in IAMR, however, we take advantage of the conservative form of the viscous stress tensor as pointed out by Balakrishnan et al.[200]. Noting that the stress divergence is

$$\nabla \cdot \boldsymbol{\tau} = \nabla \cdot (\eta \dot{\boldsymbol{\gamma}}) = \nabla \cdot (\eta \nabla \boldsymbol{u}) + \nabla \cdot \left(\eta \left(\nabla \boldsymbol{u}\right)^\top\right), \tag{7.4}$$

and thus contains one term with purely unmixed derivatives in $\boldsymbol{u}$, and one with mixed and transverse derivatives, we treat the former implicitly and the latter explicitly. This is due to the fact that the unmixed term is of a compatible layout with the temporal derivative of $\boldsymbol{u}$, and can readily be set up for a linear solve. The explicit viscous term, derived from the input velocity is therefore

$$\boldsymbol{V} = \frac{1}{\rho}\nabla \cdot \left(\eta \left(\nabla \boldsymbol{u}\right)^\top\right), \tag{7.5}$$

where η (via $\|\dot{\boldsymbol{\gamma}}\|$) is calculated from the face-centred extrapolated velocity components $\boldsymbol{u}^F$, as shown on lines 6-8 in algorithm 7.2. Note that one of the rheological constitutive equations in chapter 2.2 must be specified for the simulation.

In the predictor, the derived quantities from the input velocity $\boldsymbol{u}^m$ are used directly in the velocity update. In the corrector, on the other hand, the derived quantities are first computed from the predictor's output velocity $\boldsymbol{u}^*$, before the average is calculated and utilised in the update. See lines 6-8 and 15-17 in algorithm 7.1. The semi-implicit velocity update consists of solving the system

$$\left(\rho - \Delta t \nabla \cdot (\eta \nabla)\right) \tilde{\boldsymbol{u}} = \rho \boldsymbol{u} + \Delta t \left(\rho \boldsymbol{C} + \boldsymbol{V} - \nabla p + \boldsymbol{f}\right) \tag{7.6}$$

with respect to $\tilde{\boldsymbol{u}}$, the new-time velocity.

It is necessary to apply another projection in order to enforce the incompressibility constraint for the new velocity. Conveniently, we can update the pressure at the same time. The two equations

$$\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^{\text{old}}}{\Delta t} = -\boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho}\left(\nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}\right) \tag{7.7a}$$

$$\frac{\boldsymbol{u}^{\text{new}} - \tilde{\boldsymbol{u}}}{\Delta t} = -\frac{1}{\rho} \nabla p \tag{7.7b}$$

sum to a MOL discretised version of (2.29b), and the latter is of the form of a Helmholtz decomposition, just like (7.2). We therefore add the pressure gradient term back to the new-time velocity,

$$\tilde{\boldsymbol{u}} \leftarrow \tilde{\boldsymbol{u}} + \frac{\Delta t}{\rho} \nabla p \tag{7.8}$$

before solving the Poisson equation

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{\text{new}}\right) = \nabla \cdot \tilde{\boldsymbol{u}}, \tag{7.9}$$

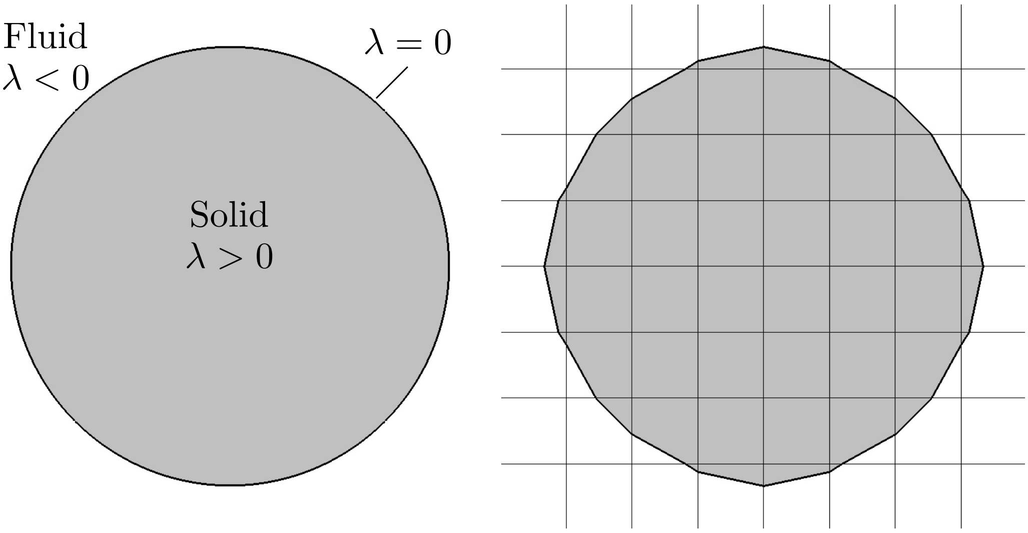
for p^{new}, and obtain the new, divergence-free velocity field

$$\boldsymbol{u}^{\text{new}} = \tilde{\boldsymbol{u}} - \frac{\Delta t}{\rho} \nabla p^{\text{new}}. \tag{7.10}$$

Note that in this projection, all velocity components are stored on cell centres, and the pressure is thus nodal. Consequently, we refer to it as the nodal projection. It is in fact a second-order accurate approximate projection method, the likes of which have been thoroughly analysed by Almgren et al.[201]. The predictor outputs the velocity field $\boldsymbol{u}^*$ and pressure p^*, while for the corrector the corresponding variables are $\boldsymbol{u}^{m+1}$ and p^{m+1}. After the corrector, Δt is added to the current time, and m is incremented, before continuing on to the next time step.

7.2 Embedded boundary treatment

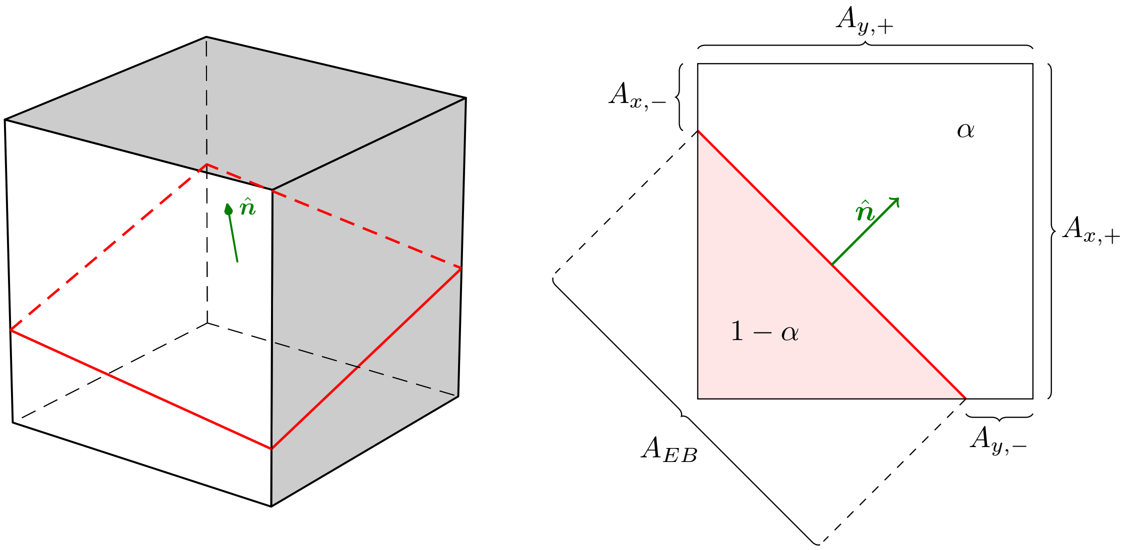
The EB approach allows us to retain the block-structured adaptive mesh which AMReX is built for, while simulating flow in non-rectangular domain boundaries. An arbitrary implicit signed distance function, commonly referred to as the level-set function, is used to describe the geometry. It is discretised as planar intersections with each cell. The intersections are continuous at cell faces, which means that they are piecewise linear everywhere. Figure 7.1 illustrates the level-set function and embedded boundary representation for a simple circle in a 8 × 8 2D grid. Cells are identified by their index vector $\boldsymbol{i} = (i, j, k)$.



(a) Level set function $\lambda(\boldsymbol{x})$ (b) Discretised grid with cut cells

Figure 7.1: Two-dimensional embedded boundary representation of a circular solid region. The example shows how a solid boundary can be represented by a spatially varying level-set function, and how this can be discretised in order to obtain a numerical mesh with planar intersections within each cut cell.

We store a flag in each cell marking it as either uncovered (normal), covered (ignored), or cut. In the latter case, additional data is stored within the cell, so that we can take geometrical information into account for computations in that cell. Since we only consider single-valued cut cells, only four numbers are necessary to uniquely define the cut cell, namely the three components of the boundary surface unit normal $\boldsymbol{n}_{EB}$ plus the volume fraction $\alpha \in (0, 1)$ of fluid within the cell. We orient $\boldsymbol{n}_{EB}$ so that it points into the fluid domain, and let $\alpha = 0$ for covered cells and $\alpha = 1$ for uncovered ones. Although these quantities describe the cut cell, several helpful additions are available in the EB framework provided by AMReX, such as the uncovered area fractions on each face, the volume centroid of the fluid, and the area and centroid of the embedded boundary. Figure 7.2 illustrates the cut cell and the relevant quantities for our computations. For details on the EB implementation, we refer to the AMReX documentation, and for an explanation of how EB data is computed from a signed distance function, the reader is referred to the paper by Gokhale et al.[198]



(a) 3D view

(b) 2D view of xy-plane

Figure 7.2: Schematic illustration of the Cartesian cut cell viewed in 3D (left) and through a slice with constant z (right). The plane representing the embedded boundary is marked in red, and its surface normal vector $\boldsymbol{n}$ points from the covered solid region to the uncovered fluid domain with volume fraction α. The right plot also shows the relevant surface areas.

All variables in our simulations are stored at the same cell positions, regardless of the cell type. This has the strong advantage of allowing us to avoid altering computational stencils for calculating extrapolated values on faces or approximating gradients based on neighbouring cells which are cut. We only need to worry about the cut cells themselves, which can have covered (and thus undefined) neighbours.

The functionality provided by the AMReX framework facilitates simple solutions to many of the complex subproblems we encounter. For example, all linear systems solved in our software utilises AMReX' built-in solvers for EB systems. The EB tools built in to the linear solvers for systems of linear equations (solutions to the Poisson equations) are based on the work of Johansen and Colella[194].

Flux computations

Our main challenge algorithmically is to successfully compute the derived quantities within each cut cell, i.e. the convective term $\boldsymbol{C}$, the rate-of-strain tensor magnitude $\|\dot{\boldsymbol{\gamma}}\|$, the apparent viscosity η and the viscous term $\boldsymbol{V}$. It is crucial that this is done in a manner which avoids time step constraints resulting from small cut cell volumes.

Let us first consider terms which can be written as the divergence of a flux $\boldsymbol{F}$, i.e.

$$\boldsymbol{D} = \nabla \cdot \boldsymbol{F}. \tag{7.11}$$

For these, we utilise the flux redistribution technique for embedded boundaries as developed by Colella et al.[193;202]. Applying the divergence theorem to a cut cell control volume, we find that

$$\boldsymbol{D}_{\boldsymbol{i}}^{C} = \frac{1}{V_{\boldsymbol{i}}} \int_{\Omega_{\boldsymbol{i}}} \nabla \cdot \boldsymbol{F} \,\mathrm{d}V = \frac{1}{V_{\boldsymbol{i}}} \int_{\partial\Omega_{\boldsymbol{i}}} \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{A} = \frac{1}{V_{\boldsymbol{i}}} \sum_{f=1}^{N_{\boldsymbol{i}}} A_f \boldsymbol{F} \cdot \boldsymbol{n}_f \tag{7.12}$$

is a conservative estimate of $\boldsymbol{D}$. Here, $V_{\boldsymbol{i}} = \alpha_{\boldsymbol{i}} \Delta x \Delta y \Delta z$ is the cell volume, $N_{\boldsymbol{i}}$ is the number of faces (EB or otherwise) enclosing the cell, and A_f is the uncovered surface area of the face f, while $\boldsymbol{n}_f$ is its normal vector. For all non-EB faces, we can evaluate the flux tensor $\boldsymbol{F}$ just as for uncovered cells, since physical variables are stored at cell centres. Consequently, the slope computation and upwinding procedures do not need to be altered for EB cells except that one-sided upwinding is applied in the case of covered neighbouring cells. In other words, we just need to use the EB information to extrapolate $\boldsymbol{F}$ to the uncovered face centroid and multiply it by the corresponding face area. Subscripting fluxes and area fractions in direction d by $+$ at one end and $-$ at the other, (7.12) can be written

$$\boldsymbol{D}_{\boldsymbol{i}}^{C} = \frac{1}{V_{\boldsymbol{i}}} \sum_{d \in \{x,y,z\}} \left(\boldsymbol{F}_{d,+} A_{d,+} - \boldsymbol{F}_{d,-} A_{d,-} \right) \cdot \hat{\boldsymbol{e}}_d + \frac{1}{V_{\boldsymbol{i}}} \boldsymbol{F}_{EB} A_{EB} \cdot \boldsymbol{n}_{EB}, \tag{7.13}$$

and the only special consideration required is the evaluation of the flux tensor at the EB centroid, $\boldsymbol{F}_{EB}$. Figure 7.3 displays these fluxes in a 2D slice of the cut cell.

The downside of the conservative flux as computed in (7.13) is that the so-called small cell problem arises in the explicit temporal discretisation[203;204]. The time step size is restricted since it is proportional to α, which can be arbitrary small in cut cells. In order to circumvent this, we define a non-conservative approximation to the divergence as an average of the conservative approximations in the neighbourhood of

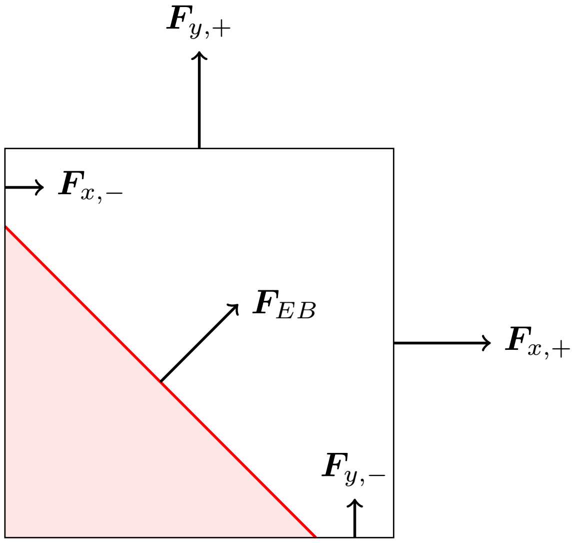


Figure 7.3: Decomposition of fluxes employed in the cell-averaged evaluation of divergence. Summation of these fluxes multiplied by cell surface areas leads to a conservative estimate of the divergence of the flux field.

the cell,

$$\boldsymbol{D}_{\boldsymbol{i}}^{NC} = \frac{\sum\limits_{\boldsymbol{i}' \in \mathcal{N}(\boldsymbol{i})} \alpha_{\boldsymbol{i}'} \boldsymbol{D}_{\boldsymbol{i}'}^{C}}{\sum\limits_{\boldsymbol{i}' \in \mathcal{N}(\boldsymbol{i})} \alpha_{\boldsymbol{i}'}}. \tag{7.14}$$

We define the neighbourhood as the set

$$\mathcal{N}(\boldsymbol{i}) = \left\{ \boldsymbol{i}' \in \mathbb{Z}^3 : \min|\boldsymbol{i} - \boldsymbol{i}'| = \max|\boldsymbol{i} - \boldsymbol{i}'| = 1, \alpha_{\boldsymbol{i}'} > 0 \right\}, \tag{7.15}$$

i.e. all uncovered or cut cells whose index vector components differ by at most one from $\boldsymbol{i}$, except cell $\boldsymbol{i}$ itself. A linear hybridisation of the conservative and non-conservative flux approximations is then given by

$$\boldsymbol{D}_{\boldsymbol{i}}^{H} = \alpha_{\boldsymbol{i}} \boldsymbol{D}_{\boldsymbol{i}}^{C} + (1 - \alpha_{\boldsymbol{i}}) \boldsymbol{D}_{\boldsymbol{i}}^{NC}. \tag{7.16}$$

Note that this hybrid flux approximation has the desired behaviour in the limits of $\alpha_{\boldsymbol{i}} \in [0, 1]$, and removes the effect of the local volume fraction on the time step size.

Although the hybrid flux stabilises the CFL restriction, it does not strictly enforce conservation. The excess material lost or gained due to the usage of (7.16) rather than (7.13) is

$$\delta \boldsymbol{D}_{\boldsymbol{i}} = \alpha_{\boldsymbol{i}} \left(\boldsymbol{D}_{\boldsymbol{i}}^{C} - \boldsymbol{D}_{\boldsymbol{i}}^{H} \right) = \alpha_{\boldsymbol{i}} (1 - \alpha_{\boldsymbol{i}}) \left(\boldsymbol{D}_{\boldsymbol{i}}^{C} - \boldsymbol{D}_{\boldsymbol{i}}^{NC} \right), \tag{7.17}$$

and this excess must be redistributed back to the cell neighbours. This is done with the use of weights $w_{\boldsymbol{i},\boldsymbol{i}'} \geq 0$ which quantify the fraction of $\delta \boldsymbol{D}_{\boldsymbol{i}}$ redistributed to cell $\boldsymbol{i}'$, and which must satisfy $\sum_{\boldsymbol{i}' \in \mathcal{N}(\boldsymbol{i})} w_{\boldsymbol{i},\boldsymbol{i}'} \alpha_{\boldsymbol{i}'} = 1$ for strict conversation. We use the simple weights

$$w_{\boldsymbol{i},\boldsymbol{i}'} = \frac{1}{\sum\limits_{\boldsymbol{i}' \in \mathcal{N}(\boldsymbol{i})} \alpha_{\boldsymbol{i}'}}, \tag{7.18}$$

which are actually independent of $\boldsymbol{i}'$. The final, discrete approximation to the divergence operator is thus

$$\boldsymbol{D}_{\boldsymbol{i}} = \boldsymbol{D}_{\boldsymbol{i}}^{H} + \sum_{\boldsymbol{i}' \in \mathcal{N}(\boldsymbol{i})} w_{\boldsymbol{i}',\boldsymbol{i}} \delta \boldsymbol{D}_{\boldsymbol{i}'}. \tag{7.19}$$

Convective term

Seeing the convective term as the divergence of a tensor flux, $\boldsymbol{C} = \nabla \cdot \boldsymbol{F}^{\boldsymbol{C}}$, we can apply the procedure outlined above in order to compute the convective term in EB cells. Note that we enforce an inhomogeneous Dirichlet condition on $\boldsymbol{u}$ at all EB walls, so that the components of the convective flux tensor are all zero there: $\boldsymbol{F}_{EB}^{\boldsymbol{C}} = \boldsymbol{u} \otimes \boldsymbol{u}^{EB} = 0$. As such, we do not need to make any special considerations for the convective fluxes at embedded boundaries. This is in contrast to the viscous wall fluxes, which we will deal with next.

Viscous term

The flux tensor arising from the viscous term is $\boldsymbol{F}^{\boldsymbol{V}} = \eta \left(\nabla \boldsymbol{u}\right)^{\top}$, which can be non-zero on the EB surfaces. We therefore need a procedure to compute the viscous wall flux at the EB surface centroid $\boldsymbol{b}$, which is given relative to local cut cell coordinates, where the cell centre is the origin. To this end, we compute the gradient of each velocity component along the EB surface normal vector $\boldsymbol{n}_{EB}$. In order to achieve this, we utilise AMReX' built-in biquadratic interpolation routine to find the value of $\boldsymbol{u}$ at two points located at distances d_1 and d_2 from the EB surface centroid $\boldsymbol{b}$. Figure 7.4 illustrates the interpolation points for an example cut cell.

We start by finding which is the largest component of the surface normal vector. The biquadratic interpolation will be done in planes where the corresponding coordinate is held fixed. Consider the case when $\max(n_x, n_y, n_z) = n_x$, as in figure 7.4. In order to make sure that we are moving away from the EB, we let $s = \text{sign}(n_x)$ and define the interpolation points as those where the line $\boldsymbol{b} + d\boldsymbol{n}$ intersect the planes $x = s$ and

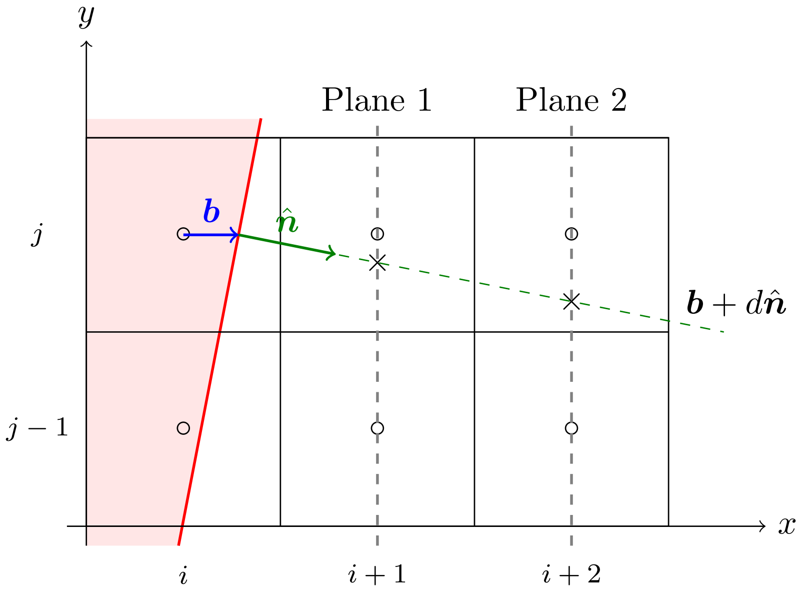


Figure 7.4: Computation of viscous wall fluxes at the EB centroid of cell i, j, k. The red line represents the embedded boundary, which cuts the cell such that its surface centroid and normal are $\boldsymbol{b}$ and $\boldsymbol{n}$, respectively. In this example, the largest component of the surface normal is in the x-direction, so we interpolate to the points marked with crosses in the yz-planes corresponding to $i+1$ and $i+2$.

$x = 2s$. The corresponding distances from $\boldsymbol{b}$ are

$$d_1 = \frac{s - b_x}{n_x}, \quad d_2 = \frac{2s - b_x}{n_x}. \tag{7.20}$$

We can thus find the y and z coordinates, and utilise biquadratic interpolation to obtain velocity values based on the 9 nearest points in the plane.

Given a set of $n+1$ real values y_i at data points x_i for $i \in \{0, \ldots, n\}$, polynomial interpolation seeks to find a polynomial curve whose curve passes through each of them. The Lagrange form of the interpolant is given by

$$f(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}, \tag{7.21}$$

so that it's derivative is

$$\frac{\partial f}{\partial x} = \sum_{i=0}^{n} y_i \frac{\sum_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} = \sum_{i=0}^{n} y_i \frac{nx - \sum_{j \neq i} x_j}{\prod_{j \neq i}(x_i - x_j)}. \tag{7.22}$$

The polynomial interpolant is unique and accurate to order n.

Denoting the interpolated velocities by $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, and allowing a prescribed velocity $\boldsymbol{u}_{EB}$ on the boundary, (7.22) gives the normal derivative as

$$\left.\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right|_{EB} = \frac{d_2^2(\boldsymbol{u}_1 - \boldsymbol{u}_{EB}) - d_1^2(\boldsymbol{u}_2 - \boldsymbol{u}_{EB})}{d_1 d_2 (d_2 - d_1)}. \tag{7.23}$$

All components of the velocity gradient are now available by taking the projections of $\partial \boldsymbol{u}/\partial \boldsymbol{n}$ in the relevant Cartesian direction, i.e. multiplying by the corresponding component of $\boldsymbol{n}$. We can thus compute $\dot{\boldsymbol{\gamma}}$, η and finally $\boldsymbol{F}^V = \eta\left(\nabla \boldsymbol{u}\right)^\top$ in the cut cell.

Strain-rate tensor and apparent viscosity

Since our aim is to accurately capture the flow patterns of fluids with apparent viscosity functions which depend strongly on the magnitude of the rate-of-strain tensor, it is essential that we can compute the components of the velocity gradient tensor to second-order accuracy. The procedure outlined for the viscous wall fluxes above is tailored for computing the values on EB walls, but in the present case we require them at cell centres. In order to circumvent the problem of covered neighbour cells, we adjust the stencil used for difference estimation. Similarly to in the previous subsection, we consider a function $y(x)$ whose values we know at three points $x_i, i \in \{0, 1, 2\}$, with the alteration that $x_0 \neq 0$ but the points have equal spacing Δx. According to (7.22), the gradient of a quadratic polynomial fit to these points is given by

$$\frac{\partial f}{\partial x} = \frac{1}{\Delta x^2}\left(\frac{1}{2}(x - x_1 + x - x_2)y_0 - (x - x_2 + x - x_0)y_1 + \frac{1}{2}(x - x_0 + x - x_1)y_2\right). \tag{7.24}$$

We can thus evaluate the gradient of the function at the given points to second order accuracy with simple coefficients, since

$$\left.\frac{\partial f}{\partial x}\right|_{x_0} = \frac{1}{\Delta x}\left(-\frac{3}{2}y_0 + 2y_1 - \frac{1}{2}y_2\right) \tag{7.25a}$$

$$\left.\frac{\partial f}{\partial x}\right|_{x_1} = \frac{1}{2\Delta x}\left(y_2 - y_0\right), \tag{7.25b}$$

$$\left.\frac{\partial f}{\partial x}\right|_{x_2} = \frac{1}{\Delta x}\left(\frac{1}{2}y_0 - 2y_1 + \frac{3}{2}y_2\right). \tag{7.25c}$$

In order to evaluate the velocity gradient tensor in cut cells, we simply check whether the neighbouring cells in each direction are covered, and if so, utilise a one-sided quadratic difference estimator rather than the central one. Consequently, we are only fishing for values in well-defined cells. With the estimates for velocity gradient components in place, we can evaluate the cell-centred rate-of-strain magnitude and apparent viscosity directly.

7.3 Validation studies

Before evaluating the methodology for genuinely three-dimensional viscoplastic flows, we need to ensure that the underlying incompressible flow solver works for a problem with non-trivial domain and known solution. Furthermore, we verify that viscoplastic effects are captured by computing the solution to a problem which has an analytical solution for the Bingham model.

Newtonian Taylor-Couette flow

To start off with, we consider the case of Taylor-Couette flow between two concentric cylinders. The setup is extensively used in rheometry, since it is straightforward to set up experimentally and gives consistent measurements of viscous drag as a function of applied strain-rate. The two cylinders can be rotating in the same or opposing directions at given speeds, and since we can derive an analytical solution for the Newtonian case, it is an ideal case for validation of prescribed tangential velocities on the no-slip boundaries. In figure 7.5, we illustrate the problem conceptually.

The inner and outer cylinders are denoted by 1 and 2, respectively. We thus take $\mathcal{R}_1$ as the radius of the inner cylinder, while Ω_2 is the angular velocity of the outer one, etc. Note that the sign of the angular velocity is positive if the rotation is counter-clockwise as viewed from above. The fluid motion is driven entirely by the rotating cylinders, so the pressure gradient is zero and the flow is purely rotational. In cylindrical coordinates, the flow field is therefore

$$\boldsymbol{u} = u_r \hat{\boldsymbol{e}}_r + u_\theta \hat{\boldsymbol{e}}_\theta + u_z \hat{\boldsymbol{e}}_z = \omega r \hat{\boldsymbol{e}}_\theta. \tag{7.26}$$

The steady-state solution is governed by

$$\nabla^2 u_\theta = 0, \tag{7.27}$$

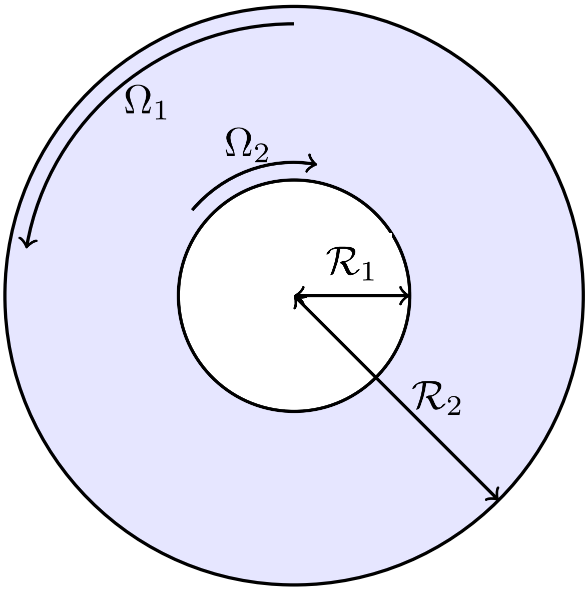


Figure 7.5: Schematic illustration of the Taylor-Couette cell, with rotational flow induced by prescribed inner and outer wall speeds.

which in terms of the angular velocity can be written

$$r\frac{\partial^2\omega}{\partial r^2} + 3\frac{\partial\omega}{\partial r} = 0. \tag{7.28}$$

The solution to (7.28) which satisfies the boundary conditions $\omega_i(\mathcal{R}_i) = \Omega_i$ is

$$\omega(r) = \frac{\Omega_1}{1-\xi^2}\left((1-\chi)\frac{\mathcal{R}_1^2}{r^2} + \chi - \xi^2\right), \tag{7.29}$$

where we have introduced the dimensionless quantities $\xi = \mathcal{R}_1/\mathcal{R}_2$ and $\chi = \Omega_2/\Omega_1$. Taking the radius and speed of the inner cylinder as characteristic values for length and velocity, the analytical solution in dimensionless form is

$$\hat{\omega}(\hat{r}) = \frac{1}{1-\xi^2}\left(\frac{1-\chi}{\hat{r}^2} + \chi - \xi^2\right). \tag{7.30}$$

We take $\xi = -\chi = 1/4$, which means that the outer cylinder is four times as wide as the inner one, and that they are rotating in opposite directions with the same speed $\mathcal{U} = -\Omega_1\mathcal{R}_1 = \Omega_2\mathcal{R}_2$. Figure 7.6 shows the velocity profile for $N = 200$ cells per direction. The fluid near the inner cylinder is rotating clockwise with maximum speed $\mathcal{U}$, while the outer one rotates at the same speed but counter-clockwise. Figure 7.7 shows that our numerical solution converges to the analytical one for increased mesh resolution. The error is greater near the inner cylinder than the outer one, which is to be expected since there is a higher degree of curvature per cell when the cell size is equal throughout the domain.

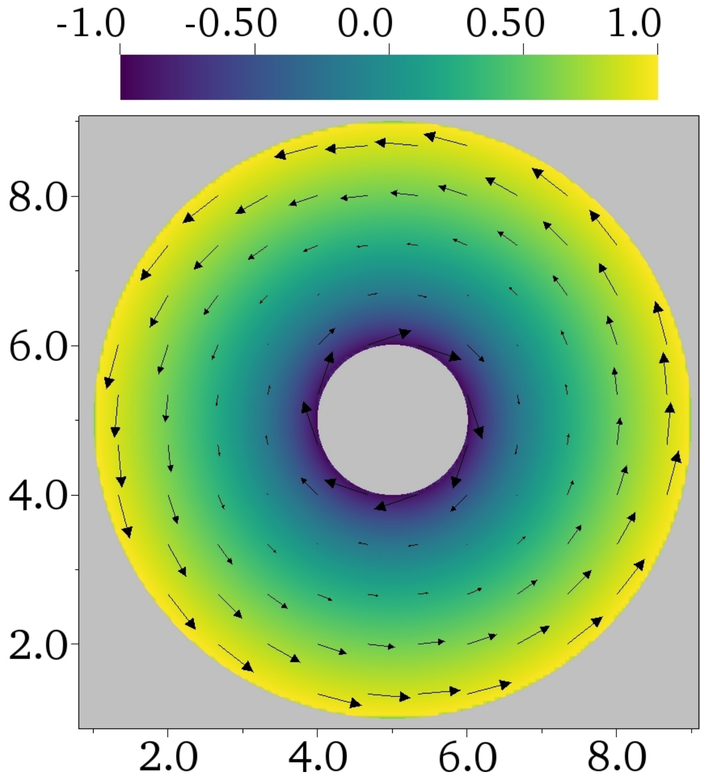


Figure 7.6: Velocity distribution in a $\hat{x}\hat{y}$-slice of the Taylor-Couette flow. The colormap corresponds to the polar velocity component $\hat{u}_\theta(\hat{r})$.

Poiseuille flow of Bingham plastic in a cylinder

In order to verify that the embedded boundaries work with viscoplastic rheology, we again consider Poiseuille flow of a Bingham fluid like in section 2.3, but now on the interior of a circular cylinder with radius $\mathcal{R}$, aligned along the z-axis. Since the problem is both axisymmetric and independent of z, the variation in velocity is a function of $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ alone, where the cylinder centre axis has coordinates (x_c, y_c, z). The analytical Bingham solution is then the cylindrical version of (2.42), i.e. with z replaced by r. Our simulations were performed in a domain of size $\Omega = [0, 2.5\mathcal{R}]^2 \times [0, \mathcal{R}]$ with the cylinder centred at $\hat{x}_c = \hat{y}_c = 1.25$. We set physical parameters of the applied pressure gradient and fluid such that $r_0 = 1/2$.

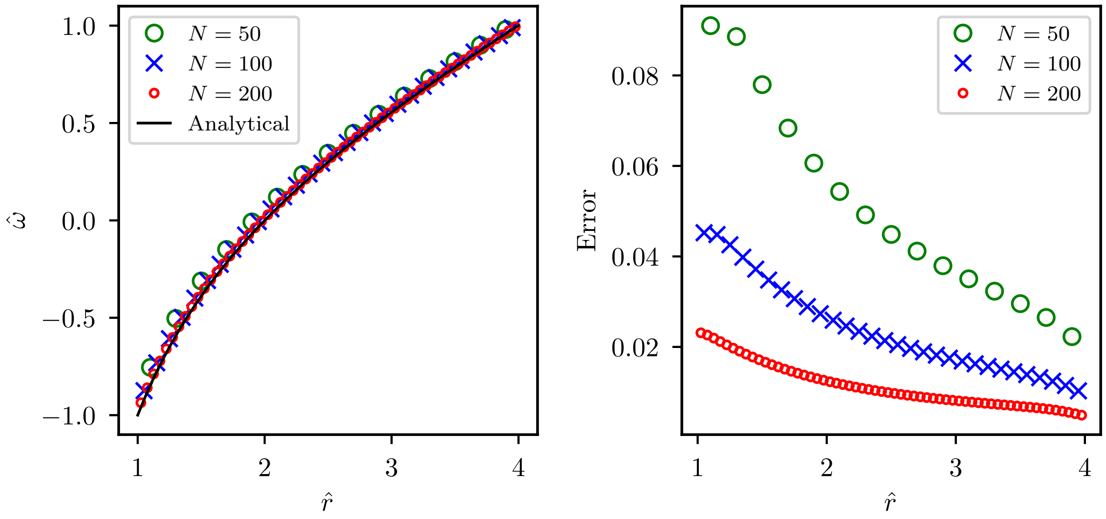
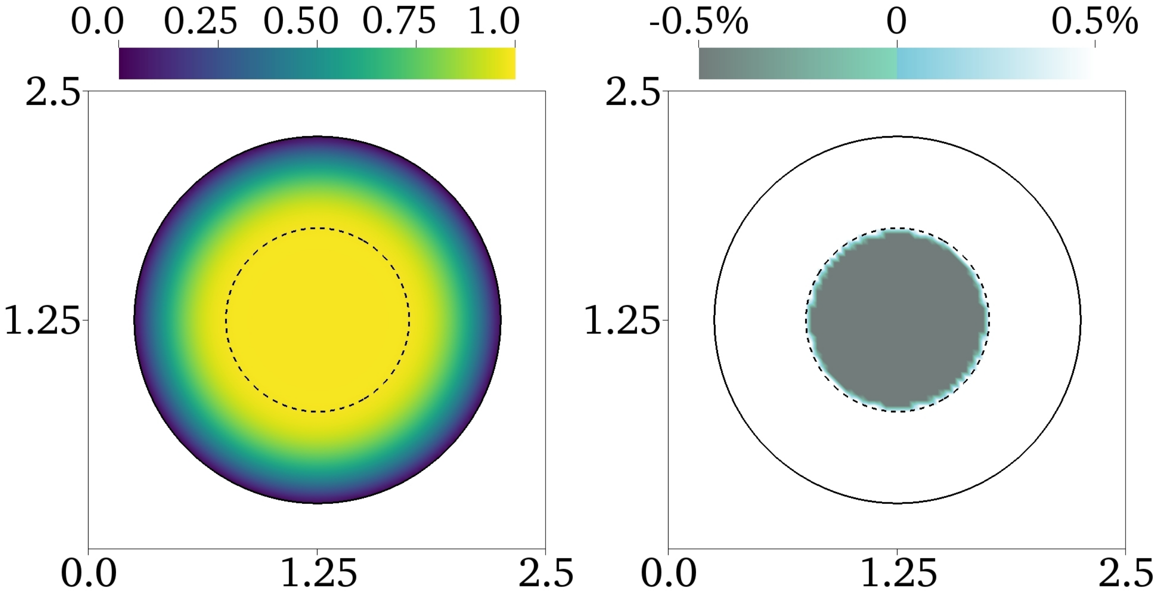


Figure 7.7: Comparison of our numerical results with the analytical angular velocity profile for Taylor-Couette flow.

Figure 7.8a shows the resulting velocity distribution in the xy-plane for a simulation with $N = 2\mathcal{R}/\Delta x = 64$ cells over the cylinder diameter, and Papanastasiou number $Pa = 100$. In the middle of the cylinder, we can see the plug of unyielded fluid travelling with constant velocity. Surrounding it, an annulus of constant shear rate leads to the parabolic profile we expect between the wall and the yield surface. The yield surface is characterised by the stress contour $\|\boldsymbol{\tau}\| = \tau_0$. However, as discussed previously in the literature, there is instability near this stress value which means that a better measure of the fully converged yield surface is the contour $\|\boldsymbol{\tau}\| = (1+\delta)\tau_0$[155], where δ is some small parameter of the order 10^{-3}. This is because the solution converges much faster in the yielded region than in the unyielded ones. On the other hand, Treskatis argues that a better visual investigation of the yield surface is obtained by plotting the relative deviation from the yield surface, $\|\boldsymbol{\tau}\|/\tau_0 - 1$, restricted to some small range around zero[104]. In this manner, we avoid the introduction of systematic error through overestimation of the unyielded regions. Note that this is done using a colormap which changes abruptly at zero, as seen in figure 7.8b.

From figure 7.8b, we can see that there is a sharp transition from the yielded to the unyielded regions which is in agreement with the analytical solution for the yield surface, $\hat{r} = r_0$. This validates that our model accurately captures the Bingham properties of the fluid, and that the regularisation parameter is small enough at $Pa = 100$ for the



(a) $\hat{w}$

(b) $\|\boldsymbol{\tau}\| / \tau_0 - 1$

Figure 7.8: Poiseuille flow in a cylinder with $r_0 = 1/2$: slices of the solution in the $\hat{x}\hat{y}$-plane. The velocity magnitude is shown in (a), while (b) depicts the relative deviation of the stress from the yield stress threshold. In the latter plot, values outside the colormap are mapped on to the endpoints. The dashed line is $\hat{r} = r_0$.

Poiseuille problem, although a smaller value will be necessary for more demanding cases.

7.4 Objects translating through Bingham media

In order to properly evaluate the capabilities of our code, we need to simulate test cases which exhibit fully three-dimensional effects which are characteristic of yield stress fluids. A widely discussed case is that of bodies moving at constant speed through a Bingham fluid. Such bodies are fully encapsulated by a so-called yield envelope, separating the unyielded bulk material from an interior recirculating flow. This interior flow has a unique topology owing to the viscoplastic nature of the fluid, and has been widely studied for two-dimensional[205–211] and axisymmetric three-dimensional[212;213] shapes.

Cylinder moving through Bingham fluid

Although creeping flow past a sphere in a viscoplastic fluid was investigated first[205], the simpler case is that of an infinitely long circular cylinder. This is due to the fact that only a thin slice in 3D is required for comparisons with two-dimensional reference results, but also because the viscoplastic effects are more dominant due to the extended geometry. Consequently, there is no ambiguity regarding the resulting shape of yield surfaces. The problem was first considered by Adachi and Yoshioka[214], who used variational principles and slip-line analyses to estimate the relevant drag effects. Randolph and Houlsby[215] used plasticity theory to investigate the same problem in the context of pile pressure in cohesive soil. Two excellent references which utilised numerical simulations to explore the problem are the papers by Mitsoulis[95] and Tokpavi et al.[209], in which the flow around a highly resolved 2D quadrant surrounding the cylinder was simulated with Papanastasiou regularisation schemes. The former investigated the influence of wall effects for creeping flow, while the latter extended the parameter ranges and analysis of the problem. They showed that the occurrence of characteristic dips in the yield envelope fore and aft of the cylinder is clearly prominent for Bingham fluids, in addition to small unyielded caps attached to the poles of the cylinder and rigidly rotating unyielded plugs in its equatorial plane. The same has been demonstrated by Chaparian and Frigaard[210] by using an augmented Lagrangian approach, which captures the yield surface without regularisation. They also show how slip line field theory captures the yield envelope and polar caps well, but is ill-suited for the equatorial plugs. It is also worth mentioning that there have been a number of studies with multiple cylinders in various collinear configurations, all showing similar large-scale features[211;212;216]. Since the problem is essentially two-dimensional due to its planar symmetry, we expect to see the same yield surface shapes as the aforementioned authors, even though our code is fully three-dimensional.

We thus consider an infinitely long circular cylinder, where the direction of flow (along the unit vector $\hat{\boldsymbol{e}}_z$) is perpendicular to the cylinder axis ($\hat{\boldsymbol{e}}_y$). The computational domain is $\Omega = [0, 4\mathcal{D}] \times [0, \mathcal{L}] \times [0, 6\mathcal{D}]$, where $\mathcal{L}$ is the length of the computational domain along the cylinder and $\mathcal{D}$ is its diameter, which we take as the characteristic length scale. In the following simulations, we have used $\mathcal{L} = \mathcal{D}/2$ and periodic boundary conditions in the y-direction. Rather than imposing a constant velocity on the cylinder, we keep it fixed and instead let $\boldsymbol{u} = \mathcal{U}\hat{\boldsymbol{e}}_z$ everywhere initially and at the domain boundaries as the simulation progresses. This means that the cylinder is falling with the same speed ($\mathcal{U}$) in the reference frame where the bulk fluid is at rest. Periodic boundary conditions are imposed in the axial direction of the cylinder.

Apart from the Papanastasiou number Pa, the flow is governed by the Reynolds number

$$Re = \frac{\rho \mathcal{U} \mathcal{D}}{\mu}, \tag{7.31}$$

and the Bingham number

$$Bi = \frac{\tau_0 \mathcal{D}}{\mu \mathcal{U}}, \tag{7.32}$$

We are interested in the creeping flow regime with a high degree of viscoplasticity, so we let $Re = 10^{-3}$. As numerical parameters, we let $N = \mathcal{D}/\Delta x = 64$ and $Pa = 10^3$. We expect these choices to be suitable for capturing the relevant flow features in this problem, based on the work of Liu et al.[208] and our own results in the next subsection.

For flow past bodies, quantitative comparisons with results from the literature are obtained through the computation of a drag coefficient, which is the ratio of the total drag force F_D on the body compared to a reference force. Following Tokpavi et al.[209], we take the reference as the product of the characteristic stress and the cross-sectional area of the cylinder $\mathcal{L}\mathcal{D}$. Denoting the boundary of the body by $\mathcal{B}$, the drag force is given by the surface integral

$$F_D = \int_{\mathcal{B}} \hat{\boldsymbol{e}}_z \cdot (\boldsymbol{\tau} - p\boldsymbol{I}) \cdot \mathrm{d}\boldsymbol{S}, \tag{7.33}$$

while the drag coefficient is

$$C_D = \frac{F_D}{\mu \mathcal{U} \mathcal{L}}. \tag{7.34}$$

The integral in (7.33) is computed using numerical quadrature. Since $\mathcal{B}$ is the union of embedded boundaries in the domain, the midpoint rule just requires extrapolation of the variables to the EB centroid of each cut cell and summing the contributions, i.e.

$$F_D \approx \sum_{\boldsymbol{i}:\alpha_{\boldsymbol{i}} \in (0,1)} A_{EB} \begin{bmatrix} \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} - p \end{bmatrix}_{\boldsymbol{b}_{\boldsymbol{i}}} \cdot \boldsymbol{n}_{EB}, \tag{7.35}$$

The most demanding terms to extrapolate are naturally the velocity gradients required for stress components, but we have already showed how to evaluate these in (7.23). In order to verify the drag force calculations, we initially perform a simulation with a Newtonian fluid, resulting in a computed drag coefficient equal to 21.8128. For

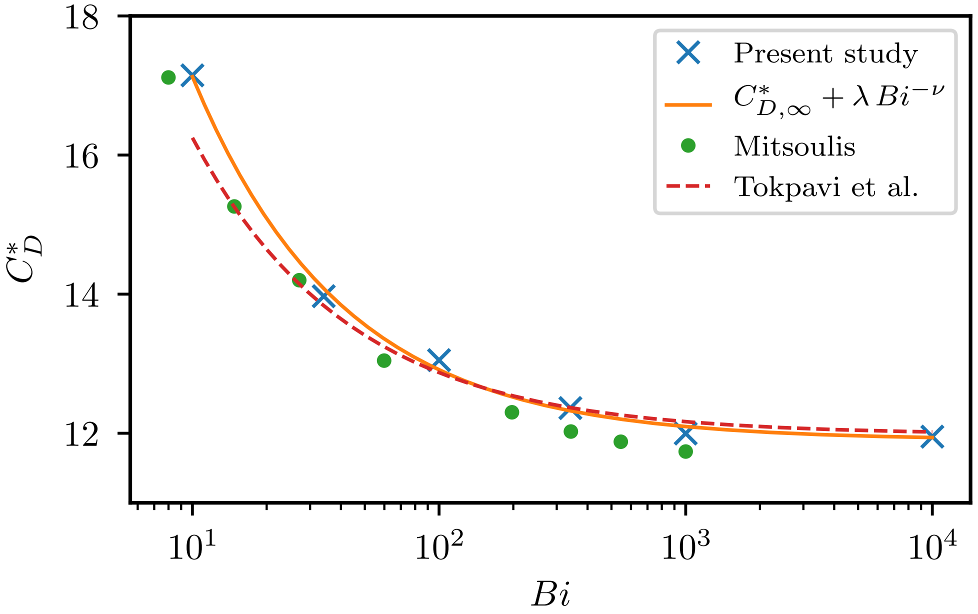


Figure 7.9: Quantitative comparison of our solution around the cylinder to that in the published literature[95;209] through computation of the drag coefficient at various Bingham numbers.

comparison, we utilise Faxén's formula[217], which states that the creeping drag force per unit length of a cylinder with diameter $\mathcal{D}$ moving at speed $\mathcal{U}$ through the centre of a channel of width $\mathcal{W}$ filled with a Newtonian fluid with viscosity coefficient μ, is given by

$$F_D = \frac{4\pi\mu\mathcal{U}}{\ln\left(\frac{\mathcal{W}}{\mathcal{D}}\right) - 0.9157 + 1.7244\left(\frac{\mathcal{D}}{\mathcal{W}}\right)^2 - 1.7302\left(\frac{\mathcal{D}}{\mathcal{W}}\right)^4 + 2.4056\left(\frac{\mathcal{D}}{\mathcal{W}}\right)^6 - 4.5913\left(\frac{\mathcal{D}}{\mathcal{W}}\right)^8}. \tag{7.36}$$

This semi-analytical formula for the drag coefficient yields a value of 21.9642, i.e. a relative discrepancy of 0.69%. By driving the system to steady-state for various Bingham numbers, we can compare our results with others which have been published in the literature. Figure 7.9 shows how the ratio $C_D^* = C_D/Bi$ gradually approaches the asymptotic value $C_{D,\infty}^*$ as Bi grows large. The numerical results of Mitsoulis[95] and Tokpavi et al.[209] are overlain our own, and the behaviour similar. Our results are closest to the trend line by Tokpavi et al., which they fit to the function

$$C_D^* = C_{D,\infty}^* + \lambda Bi^{-\nu}. \tag{7.37}$$

Our data has been fit to the same function, and we have also computed the gravity yield number

$$Y = \frac{\tau_0}{\frac{1}{4}\pi\mathcal{D}g\Delta\rho} \tag{7.38}$$

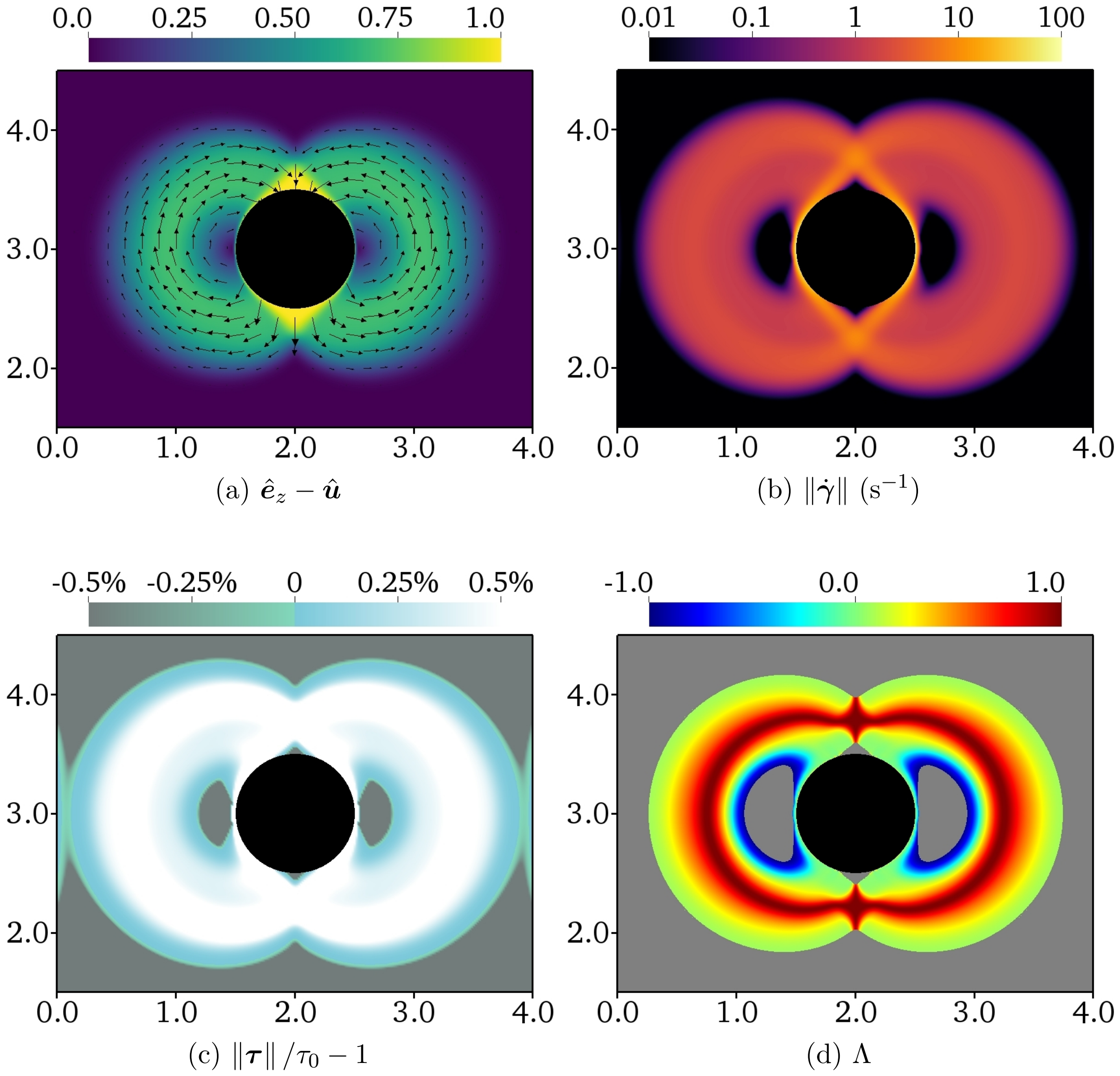
where g is the gravitational acceleration and $\Delta\rho$ is the density difference between the solid cylinder and surrounding fluid. This dimensionless number is the smallest ratio of yield stress to gravitational stresses which obstructs the solid from falling through the fluid, and for the particular configuration at hand, we have $Y = 1/C^*_{D,\infty}$[95]. Table 7.1 shows our quantitative results alongside those by other authors.

Reference	$C^*_{D,\infty}$	Y	λ	ν
Adachi & Yoshioka[214]	10.28	0.09728		
Randolph & Houlsby[215]	11.94	0.08375		
Mitsoulis[95]	11.70	0.08547		
Tokpavi et al.[209]	11.98	0.08347	20.43	0.68
Present study	11.90	0.08403	26.99	0.7131

Table 7.1: Comparison of the yield number and limiting ratio of drag coefficient to Bingham number with various reference results from the literature.

In addition to computing drag coefficients, there is a lot of insight to be gained from visualising different metrics of the flow field. Figure 7.10 therefore illuminates the resulting flow field through plots of several different variables, for the case with $Bi = 340.7$. The Bingham number is chosen to match that in the study by Liu, Muller and Denn[208], where it is stated that this number is large enough to diminish any outer boundary wall effects for the given choice of domain size. The upper left plot, (a), shows the relative speed with velocity vectors overlaid. A distinct boundary is visible, enclosing a region of non-zero relative velocity surrounding the cylinder. This boundary represents the yield envelope of the body. The velocity vectors indicate a recirculating flow, sweeping material away from the front of the travelling cylinder to the rear in a wide, circular arc. This results in slowly moving material either side of the cylinder, in addition to material travelling at the same speed as the cylinder at the polar caps. These observations are indicative of the expected unyielded plugs rotating at the equator and clinging to the polar caps.

In the upper right plot, (b), the magnitude of the rate-of-strain tensor is shown with logarithmic scaling for the colormap. This plot properly elucidates the low-strain regions of unyielded fluid, and confirm the existence of rigidly rotating equatorial



(a) $\hat{\boldsymbol{e}}_z - \hat{\boldsymbol{u}}$

(b) $\|\dot{\gamma}\|$ (s^{-1})

(c) $\|\boldsymbol{\tau}\| / \tau_0 - 1$

(d) Λ

Figure 7.10: Flow characteristics in the $\hat{x}\hat{z}$-plane through the centre of the cylinder (black) at steady-state, for $Bi = 340.7$. Depicted are (a) the relative velocity field and its magnitude, (b) the rate-of-strain magnitude, (c) the yield surface stress deviation and (d) the flow topology parameter. Note that the colormap in (b) has logarithmic scaling, and that values outside the colormap for (c) are mapped on to the endpoints. In (d), the yield surface computed with $\delta = 10^{-3}$ is also masked out in grey.

plugs and unyielded material attached to the polar caps, as implied by the velocity distribution. Note that although the yield surface is characterised by the contour $\|\dot{\gamma}\| = 0$ for ideal Bingham plastics as given by (2.26), the regularised version (3.24) leads to a yield surface given by

$$\left\|\hat{\dot{\gamma}}\right\| = \frac{W\left(Bi\, Pa\right)}{Pa}, \tag{7.39}$$

where we recall that $W(x)$ is the Lambert W function. With the current parameter values, (7.39) gives a yield surface characterised by $\|\dot{\gamma}\| = 0.010397\ \mathrm{s}^{-1}$, which is why we have used $10^{-2}\ \mathrm{s}^{-1}$ as the lower limit of the colormap.

Figure 7.10c shows the relative stress deviation from the yield surface, just like figure 7.8b did for the Poiseuille case. For this more demanding problem, the transition is much less sharp, but all the expected unyielded regions (envelope, rotating plugs and polar caps) are clearly captured where the stress deviation is negative. The actual yield surface is located somewhere in the transition from blue to white.

Finally, in figure 7.10d we have plotted a flow topology parameter which is a normalised invariant of the velocity gradient tensor. It indicates the relative strengths of the strain rate and the vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ [218;219],

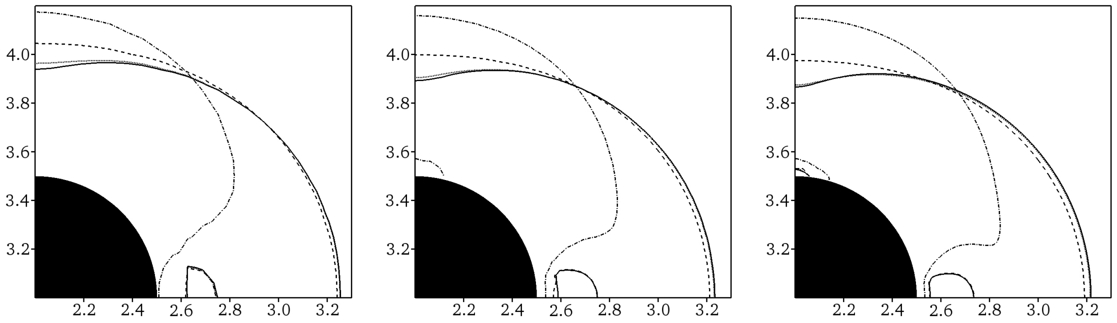
$$\Lambda = \frac{\|\dot{\gamma}\|^2 - \|\boldsymbol{\omega}\|^2}{\|\dot{\gamma}\|^2 + \|\boldsymbol{\omega}\|^2}. \tag{7.40}$$

This parameter accurately describes the nature of different parts of the flow regime, since values of Λ equal to -1, 0 and 1 correspond to pure rotation, shear and extension, respectively. As expected, we observe pure rotation around the unyielded plugs on the equatorial line of the cylinder. Adjacent to areas of rigid rotation, and near the cylinder, shear flow is evident. This is also the case near the yield envelope. In fact, these two shearing regions represent two distinct cases of viscoplastic boundary layers, as discussed originally by Oldroyd in 1947[220] and more recently by Balmforth in his newly published lecture notes[221]. We also point the interested reader to discussions surrounding viscoplastic boundary layer around particles in some of the other works which have already been mentioned[206;209;210]. Finally, a belt of purely extensional flow surrounds the cylinder, rotating plugs and polar caps. Note that in this plot, the yield surfaces masked out in grey are computed with $\delta = 10^{-3}$. This allows us to verify that we recover the same yield surface shapes as those computed in the references[209;210], when using the same visualisation procedure.

Sphere moving through Bingham fluid

Since the cylinder test case allows for direct comparisons with two-dimensional simulations, it is a good test case for verifying the interplay between yield stress rheology and the embedded boundaries. On the other hand, it does not exhibit any genuinely three-dimensional effects, and for many real-world scenarios an infinite cylinder is not a realistic representation of the actual bluff bodies. We therefore investigate the flow around a sphere in the same configuration, retaining its diameter $\mathcal{D}$ as the characteristic length scale, but extending the domain to $[0, 4\mathcal{D}]$ in the y-direction. This problem has been analysed by several authors previously[205–208;212;213], but it still warrants further attention due to gaps relating to important viscoplastic flow features. In particular, claims about how specific parts of the yield surface depend on spatial resolution and regularisation parameter are not consistent. Furthermore, few of them were based on three-dimensional simulations, and those that were tended to be hampered by low spatial resolution, making it difficult to draw any conclusions regarding the shape and extent of the yield surface.

Ansley and Smith first considered the problem in 1967, and proposed a yield surface based on slip line field theory. This included dips in the yield envelope and polar caps[205]. Although it was a great contribution, they conceded that the qualitative shape of the yield surface could not be supported by direct evidence. Beris et al. phrased the problem as a free boundary problem at the yield surface, and performed 2D simulations with a priori estimates of its location[206]. They captured the envelope dips and polar caps, and reported an equatorial torus-shaped region with low strain-rate. Its motion was described as close to the sum of rigid translation in the direction of flow and solid body rotation, but they aptly noted that perfectly rigid rotation is not possible in the configuration, except exactly in the equatorial plane. This is due to the rotated strain field around the equator of the sphere. Liu et al.[208] computed 2D results in a quadrant using Papanastasiou regularisation and finite elements on a body fitted mesh. They reported the existence of both polar caps and a equatorial torus, although the latter is unphysical according to the arguments put forth by Beris et al. They showed that the plug regions shrank with decreasing regularisation parameter, but could not infer the limiting behaviour. More recently, three-dimensional simulations in the framework of lattice Boltzmann techniques have been published, utilising a dual viscosity model[212] and Papanastasiou regularisation[213]. However, the resolution for these simulations was low ($N = 12$), so it is difficult to draw accurate conclusions regarding the yield surface shapes.



(a) $N = 16$ (b) $N = 32$ (c) $N = 64$

Figure 7.11: Contour lines of the steady-state yield surface $\|\boldsymbol{\tau}\| = 1.001\,\tau_0$ in the upper right corner of the $\hat{x}\hat{z}$-plane $\hat{y} = 2$, for a sphere translating in an infinite Bingham medium with $Bi = 340.7$. The resolution of the numerical mesh employed in the simulations increases from (a) to (c). For each resolution, the different lines correspond to the following regularisation parameter values:
-·-·-· $Pa = 10$ - - - - $Pa = 10^2$ ········ $Pa = 10^3$ —— $Pa = 10^4$

The two parameters which determine the yield surface for a given Bingham number, are the Papanastasiou number Pa and the spatial resolution as given by N. Similarly to what was done in the paper by Liu et al.[208], in Figure 7.11 we plot the computed yield surfaces in the upper right corner of the plane $\hat{y} = 2$, which goes directly through the sphere, for a range of Pa values. However, we also consider three different mesh resolutions, in order to separate the effects of the two parameters. It is clear that both parameters must have acceptable values in order for the solution to converge as expected. Without a large enough Papanastasiou number (at least $Pa = 10^3$), the yield envelope dips and low-strain equatorial plugs are not captured, whereas a low spatial resolution results in unresolved polar caps. We note that the unphysical torus-shaped yield surface at the equator is apparent also in our simulations, but this is due to the utilisation of $\delta = 10^{-3}$ in the visualisation of the yield surface, as explained below. In the remainder of this section, we let $N = 64$ and $Pa = 10^3$.

For the case of a sphere, the reference force for the computation of the drag coefficient is the Stokes force[222] that acts on a sphere falling in an infinite Newtonian medium. The Stokes drag coefficient is thus

$$C_S = \frac{F_D}{6\pi\mu\mathcal{U}\mathcal{R}}. \tag{7.41}$$

Figure 7.12 shows our results alongside previously published data in the literature. We have included the original computational results by Beris et al.[206] and Blackery and Mitsoulis[207] as dashed and dotted lines, respectively, and have also overlaid the experimental results of Ansley and Smith[205]. As is evident, our computed values for the Stokes drag coefficient lie within the same range as the references, and match the results of Beris et al. most closely.

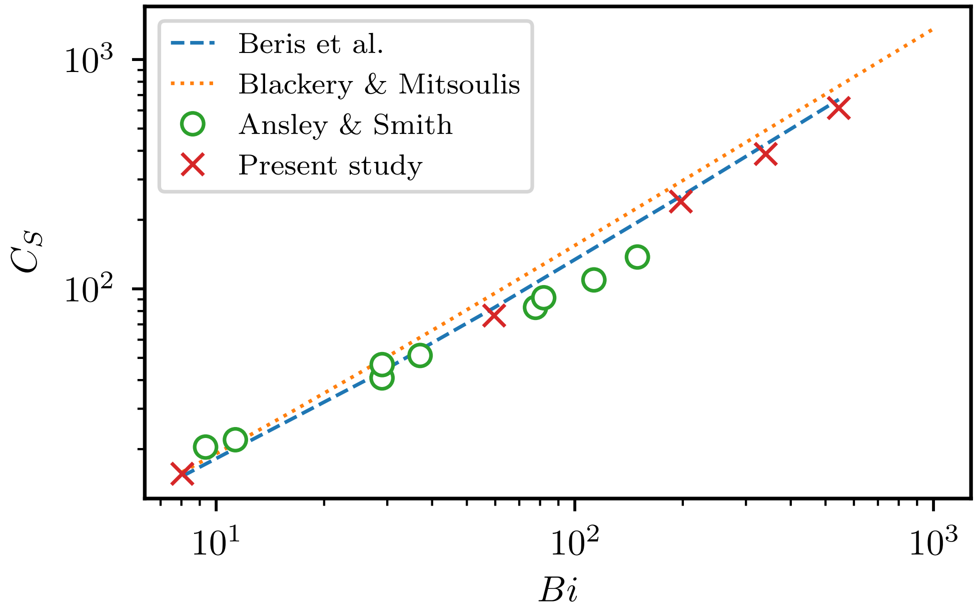
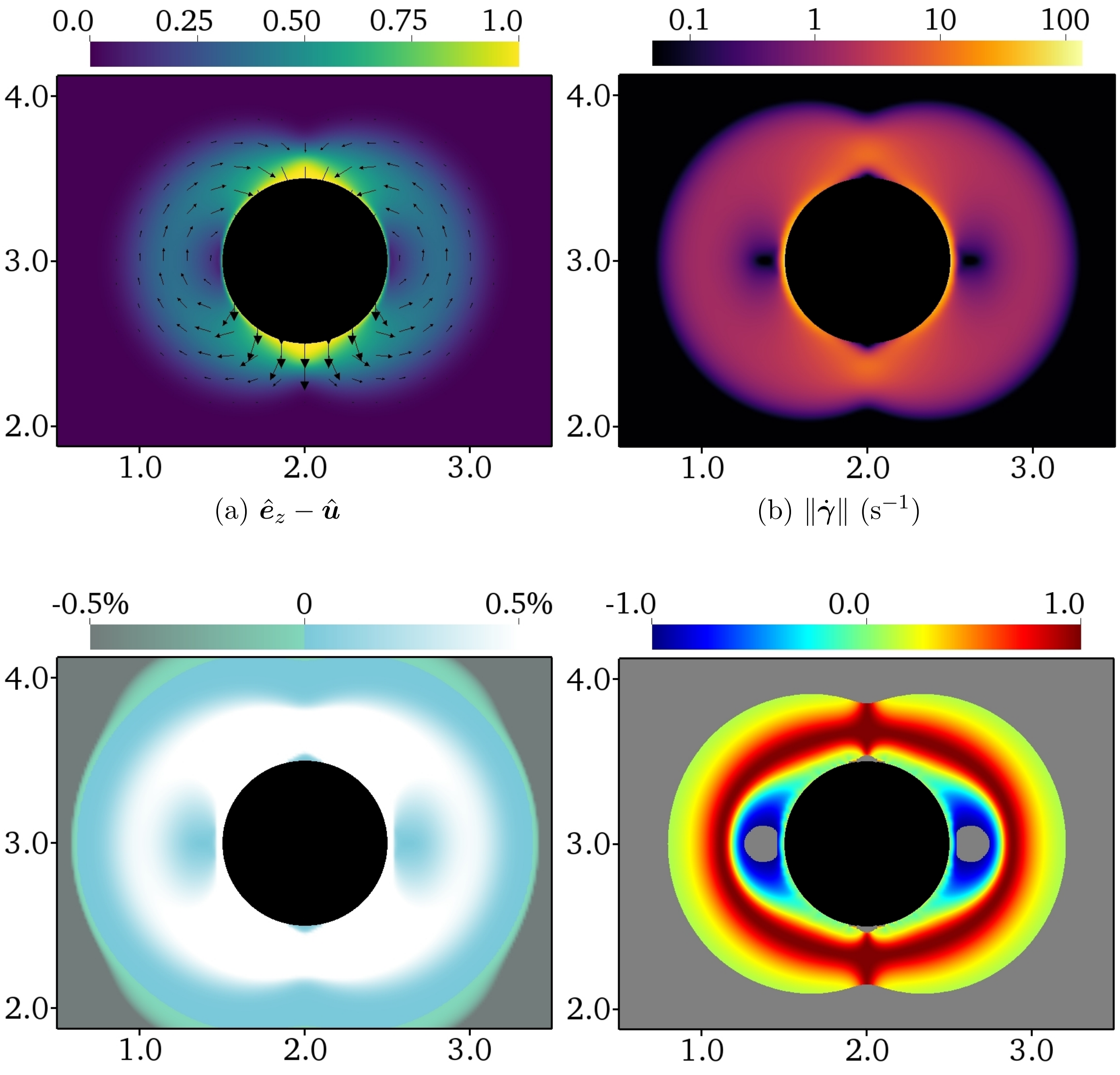


Figure 7.12: Quantitative comparison of our solution around the sphere to that in the published literature[205–207] through computation of the Stokes drag coefficient at various Bingham numbers.

In figure 7.13, we show plots of the same variables as those in figure 7.10 for the cylinder. There are several noteworthy points to make about the differences between the two cases. Firstly, the overall size of the yield envelope is significantly smaller for the sphere than the cylinder, both in the direction of flow and in the xy-plane. Secondly, the unyielded regions within the envelope are much smaller. Both of these observations are as expected, since the axial symmetry of the sphere causes reduced shearing effects in the plane. Equivalently, the extended cylinder enhances shearing in the plane, leading to the larger unyielded regions and genuine rigid body rotation. Finally, the relative stress deviation plot (c) proves that there aren't any regions in the equatorial torus where the stress is actually below the yield stress threshold. Although the stress gets extremely close to the yield threshold in these areas of low strain, there is no evidence of a genuinely unyielded region, since the relative stress plot stays blue.



(a) $\hat{\boldsymbol{e}}_z - \hat{\boldsymbol{u}}$

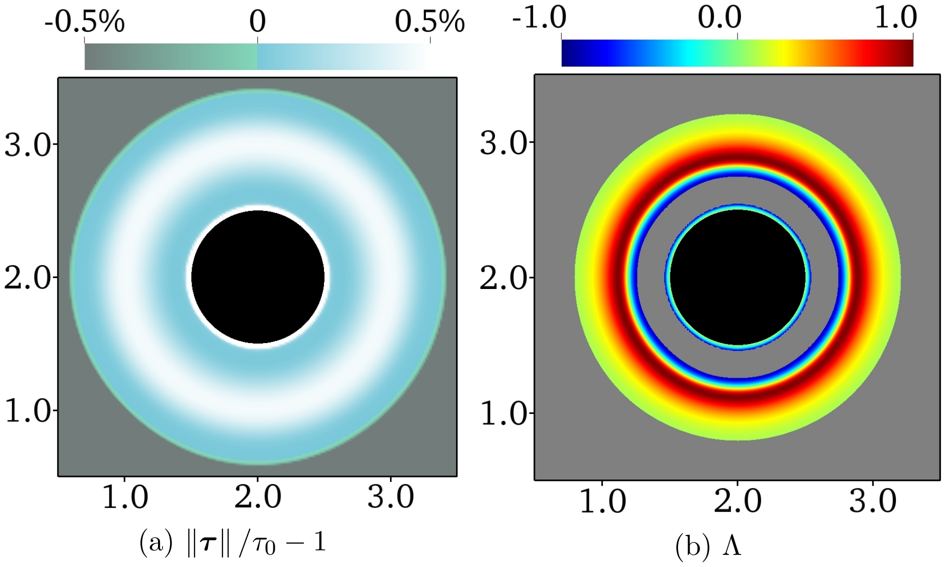
(b) $\|\dot{\boldsymbol{\gamma}}\|$ (s^{-1})

(c) $\|\boldsymbol{\tau}\| / \tau_0 - 1$

(d) Λ

Figure 7.13: Steady-state flow characteristics in the plane $\hat{y} = 2$ through the centre of the sphere in translational motion through a Bingham fluid with $Bi = 340.7$.

The diffuse interfaces in this plot shed some light on why the torus-shaped plug is present in some papers and not others: it is severely close to the threshold value without actually reaching it, and without a sharp transition near it. Thus, the benefits of the visualisation strategy introduced by Treskatis becomes clear: the plot in (c) shows in details what the stress deviation looks like near the yield surface, while the well-known masked out contour $\|\boldsymbol{\tau}\| = 1.001\ \tau_0$ in (d) oversimplifies the case. This is all in agreement with the original arguments put forth by Beris et al.[206] Figure 7.14 shows the last two plots through the slice $\hat{z} = 2$, i.e. perpendicular to the flow direction, but still through the centre of the sphere.

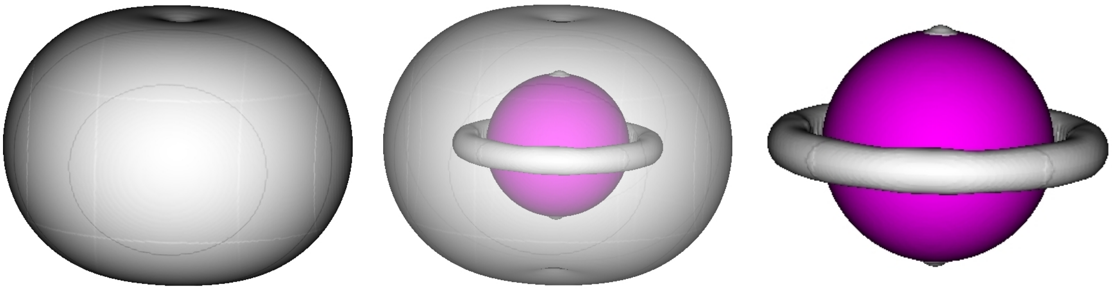


(a) $\|\boldsymbol{\tau}\|/\tau_0 - 1$

(b) Λ

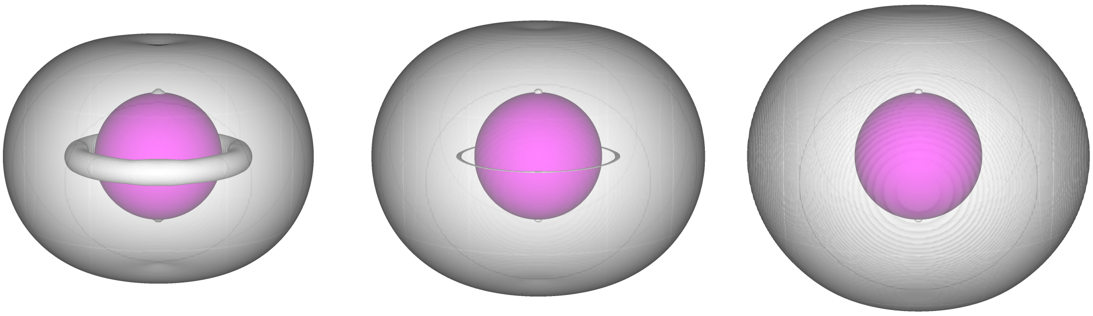
Figure 7.14: Steady-state flow characteristics in the plane $\hat{z} = 2$ through the centre of the sphere in translational motion through a Bingham fluid with $Bi = 340.7$.

Finally, in figures 7.15 and 7.16, we visualise three-dimensional stress contours in the vicinity of the sphere. In the former, the yield surfaces are all computed with $\delta = 10^{-3}$. Since the yield envelope fully encloses the sphere and plug regions, its opacity is reduced from subfigures (a) to (c), in order to reveal the shape of the enclosed yield surfaces. This kind of visualisation is entirely novel to the best of our knowledge, and gives a richer picture of the yield surface topology. Since there is no body-fitted mesh or assumption on symmetry in the flow, this type of simulation opens a range of possibilities for investigating flow patterns and yield surfaces in complex configurations. The effect of reducing δ is illustrated in figure 7.16, and confirms that the toroidal yield surface at the equator disappears in the limit $\delta \to 0$, at the cost of poorer resolution for the yield envelope.



(a) Opacity 100% (b) Opacity 50% (c) Opacity 0%

Figure 7.15: Yield surface represented by the contour $\|\boldsymbol{\tau}\| = 1.001\tau_0$ (light grey) around sphere (magenta) moving at constant speed through a Bingham fluid with $Bi = 340.7$. The opacity of the yield envelope is reduced from (a) 100% to (b) 50% and finally (c) 0%, in order to reveal the sphere and unyielded regions within it.

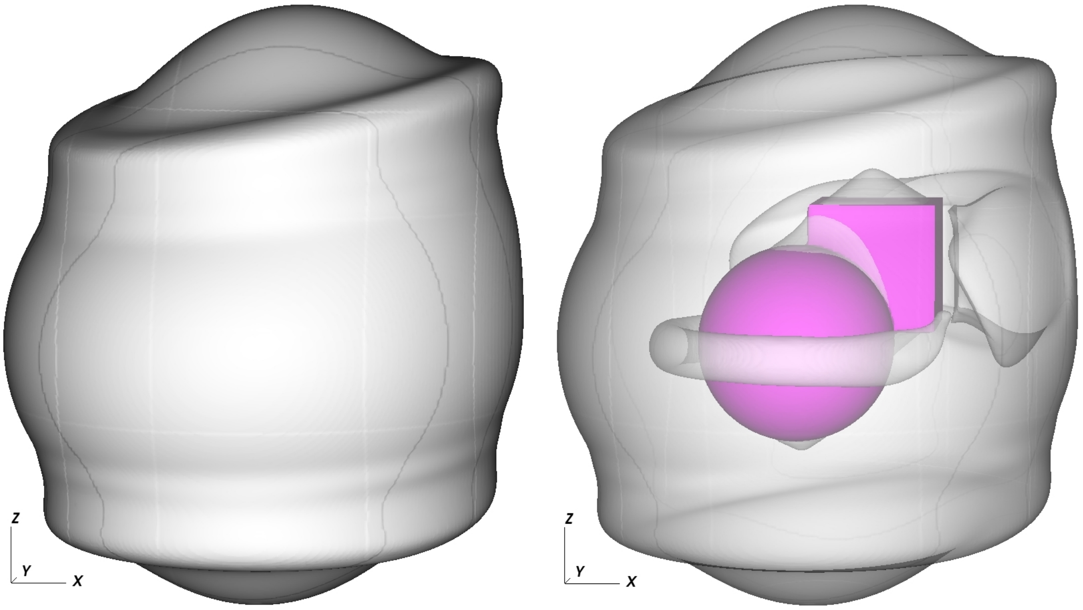


(a) $\delta = 10^{-3}$ (b) $\delta = 10^{-4}$ (c) $\delta = 0$

Figure 7.16: Yield surface represented by the contour $\|\boldsymbol{\tau}\| = (1 + \delta)\tau_0$ (light grey) around sphere (magenta) moving at constant speed through a Bingham fluid with $Bi = 340.7$. The effect of varying δ is evident, as it is reduced from (a) 10^{-3} to (b) 10^{-4} and finally (c) 0.

Non-trivial particle shape

As a final demonstration, we replace the sphere by an object which is the union of a sphere and cube, and orient it so that the flow field becomes asymmetrical. This means that a three-dimensional representation is necessary to capture the fluid dynamics. To be precise, the sphere is the same size as in the previous section, but has centre coordinates $(\hat{x}, \hat{y}, \hat{z}) = (1.8, 1.8, 2.8)$, while the cube is defined by two points on opposing sides of its main diagonal, in dimensionless units $(1.85, 1.85, 2.85)$ and $(2.5, 2.5, 3.5)$. We retain the highest resolution settings used for the sphere, i.e. $\Omega = [0, 4\mathcal{D}]^2 \times [0, 6\mathcal{D}]$, $\Delta x = \mathcal{D}/64$ and $Pa = 10^3$. The resulting 3D yield surface, computed with $\delta = 10^{-3}$, is shown in figure 7.17. In contrast to figure 7.15, we reduce the opacity of all the yield surfaces, and not just the enclosing envelope. This is due to the difficulty in separating the locations of the yield surface types by simple rules. Although the stress contours are now much more complex, we can still recognise the expected traits: an enclosing yield envelope surrounding the entire body, in addition to smaller unyielded plugs attached to it at places of low strain rate. In particular, there are caps of unyielded material fore and aft of the object in the flow direction, as well as along the narrow intersection of the sphere and cube. Additionally, the characteristic low-strain rotating region around the sphere's equator is visible, but it does not continue around the entire object. The free cube sides, which are aligned with the flow direction, only lead to a very narrow boundary layer separating the object from the yield envelope.



(a) Opacity 100% (b) Opacity 50%

Figure 7.17: Yield surface computed with $\delta = 10^{-3}$ for the flow of a Bingham fluid around the union of a sphere and a cube with $Bi = 340.7$. The opacities of the stress contours are (a) 100% and (b) 50%, in order to show unyielded regions enclosed by the outer yield envelope.

Chapter 8

Conclusions

Motivated by the emerging need for detailed understanding of displacement flows during wellbore cementing, the aim of this PhD project has been to facilitate large-scale simulations of viscoplastic fluid flows in complex domains and around non-trivial geometries, with emphasis on high computational accuracy and rapid run times. To this end, we have taken advantage of state-of-the-art techniques in high-performance computing, including massive parallelisation for arbitrary computing platforms; block-structured adaptive mesh refinement with subcycling; and efficient Krylov solvers for linear systems. These tools have been accessed through the versatile software framework AMReX, which is developed at the Lawrence Berkeley National Laboratory and among the very best in class of its field. By applying the powerful numerical tools to viscoplastic flow problems, which are burdened by severe computational challenges, we have enabled larger and more complex simulations of such fluids than previously published in the literature.

In particular, we demonstrated fully three-dimensional simulations, which vary in time, for generalised Newtonian constitutive relations such as the Herschel-Bulkley model. In order to simulate viscoplastic fluids with nonlinear stress-strain dependency above the yield limit, we utilised Papanastasiou regularisation in our algorithm to deal with the singularity in apparent viscosity. The resulting system of partial differential equations was solved using second-order Godunov methodology for the advective terms, and semi-implicit diffusion in the context of an approximate projection method for viscous effects, all solved on a hierarchy of adaptively refined meshes. We validated results from simulations using this new capability against previously published data for Bingham plastics and power-law fluids in the two-dimensional lid-driven cavity. In doing so, we expanded the range of Bingham and Reynolds numbers which have been considered in the benchmark tests. Moreover, extensions to time-dependent flow, Herschel-Bulkley fluids and three spatial dimensions offered new insights into the flow of viscoplastic fluids in this test case. Importantly, we provided missing benchmark results for these extensions[18].

As a natural continuation of that work, we extended the software using the volume of fluid method, so as to facilitate the simultaneous simulation of several interacting fluids. By passively advecting a scalar indicator with the flow velocity, the presence of each fluid throughout the domain can be ascertained, and relevant fluid properties utilised. This capability is vital in order to deal with the displacement phenomenon which characterises wellbore cementing. In addition to the validation of the method through comparisons with existing simulations results for the Rayleigh-Taylor instability, we considered the displacement of a Bingham fluid by a Newtonian, with fluid properties relevant for realistic cement jobs[19].

During the last year of the PhD, starting with a month visiting the AMReX developers in Berkeley, California, we focused on the implementation of cut cell techniques for the viscoplastic flow solver. In this context, geometric features are handled by the embedded boundary approach, which requires specialised treatment only in cells intersecting or adjacent to the boundary. This constitutes the first published implementation of an embedded boundary algorithm for simulating flow of viscoplastic fluids on structured grids. We augmented the algorithm to deal with the variable apparent viscosity of the fluids. Since the viscosity depends strongly on the strain-rate tensor, special care was taken to approximate the components of the velocity gradients robustly near boundary cells, both for viscous wall fluxes in cut cells and for updates of apparent viscosity in cells adjacent to them. After performing convergence analyses and validating the code against standard test cases, we presented the first ever fully three-dimensional simulations of creeping flow of Bingham plastics around translating objects. Our results revealed new insights into the relevant flow fields[20].

By taking advantage of some of the most advanced numerical tools available, we have demonstrated that researchers in the field of viscoplastic fluid simulations can move towards richer and more complicated flow scenarios than have previously been considered viable, within reasonable computing time. While implementing these numerical schemes and using them to produce publication-worthy fluid dynamics research, we have kept our focus on addressing the challenges met by BP while drilling and cementing wellbores. The most important features for simulating such scenarios are captured by the software, such as yield stress rheology, two-fluid displacement and complex domain geometries. We hope this software can contribute to an even better understanding of the intricacies of such flow scenarios, to ensure safer and more efficient engineering processes.

8.1 Proposed further work

Within any research and development project, there are always possibilities for extensions and improvements, so we take this opportunity to give our recommendation for future directions to follow.

Wellbore cementing case study

The most obvious piece of work which we would have pursued given more time, is to apply the software to actual wellbore cementing problems such as those introduced in chapter 1. Given the different simulation features which we have enabled through the work undertaken in this project, it would be feasible to set up and simulate a range of relevant test cases which have been studied by others using CFD tools. This would allow us to compare both accuracy and simulation speed for typical problems faced by wellbore engineers. It is not certain that our inherent scalability and use of structured meshes would outperform CFD codes with body-fitted meshes for two straight eccentric annuli, but the flexibility offered by our approach should become evident when considering e.g. bent pipes and non-circular borehole geometries. Moreover, it would have been very exciting to compare simulation results with experimental data, and start connecting physical measurements to the numerical results.

GPU computing

Over the course of the last decade, general purpose programming for Graphical Processing Units (GPUs) has become hugely popular. This is in part due to the inherent parallel processing pipeline of such devices – they work in a lock-step pattern which facilitates massive parallelisation within a single GPU card, provided the problem can be divided into equal subproblems while properly managing memory. Additionally, the programming effort required to do so has reduced significantly with the increased popularity and development of major programming language interfaces such as OpenGL and CUDA. For any computationally expensive problem, it has become natural to investigate whether GPUs can be utilised for increased efficiency. In fact, one of the latest features which the AMReX developers have been working towards for the code framework is built-in support for straightforward GPU compatibility. Taking advantage of this, and running simulations on contemporary graphics clusters could potentially speed up the run times significantly.

Temperature dependency

All of the work in this dissertation has assumed that the flow is isothermal. Although the effects of temperature are smaller than e.g. viscoplasticity, this is still an overly restrictive assumption, and one which it would be natural to dismiss when developing the model further. In order to capture the effect of temperature and heat transfer in the problem, it would be necessary to solve a temperature-dependent energy conservation equation in addition to (2.29). The temperature would then be coupled non-linearly to the remaining governing equations through temperature-dependent physical properties such as density, dynamic viscosity coefficient and yield stress. Experimental data such as that depicted in chapter 1 is ideal for fitting the relevant parameters.

Advanced boundary treatment

Although the embedded boundary approach facilitates simulations in domains with non-trivial geometries, the current implementation is restricted to no-slip walls with prescribed velocities for static boundaries. For slow, viscoplastic flows, better physical models for wall effects, such as stick-slip conditions, have been derived. Inclusion of such boundary conditions in the software would result in a broader coverage of flow simulations. Furthermore, the prescribed wall velocities which have been implemented thus far do not allow the boundary to move relative to the computational grid – this would require something like an immersed boundary method, in which the moving (and possibly deforming) structure affects the fluid through the addition of a local forcing term in the momentum equation. Immersed boundary methods for interactions between fluids and structures is a large research field in its own right, and it would be great to have this kind of capabilities available.

Fluid complexity

Generalised Newtonian fluid models allow us to capture a wide range of non-Newtonian behaviour, such as shear thinning and thickening, yield stresses and even explicit time-dependence. There are, however, many other types of non-Newtonian phenomena which require more sophisticated mathematical descriptions. This is because the non-Newtonian effects are not necessarily purely viscous. Generalised Newtonians have an apparent viscosity which only depend on the magnitude of rate of strain tensor, i.e. its second invariant, since this is the only invariant which is non-zero for shear flows. A much wider family of non-Newtonian fluids, which can capture elastic and plastic effects in addition to extensional effects due to normal stresses, are collectively specified through an extra stress tensor $\boldsymbol{\tau}_E$, related to the deviatoric stress by

$$\boldsymbol{\tau} = \boldsymbol{\tau}_E - \frac{1}{3}\mathrm{tr}(\boldsymbol{\tau}_E)\boldsymbol{I}. \tag{8.1}$$

In addition to extending the code to deal with more types of fluids through such rheological equations, it is possible to implement much more advanced interactions across fluid interfaces, such as miscibility and chemical reactivity.

References

[1] M. Couturier, D. Guillot, H. Hendriks, F. Callet, Design rules and associated spacer properties for optimal mud removal in eccentric annuli, in: CIM/SPE International Technical Meeting, Society of Petroleum Engineers, 1990, pp. 1–8.

[2] R. J. Crook, G. Benge, R. Faul, R. R. Jones, Eight steps ensure successful cement jobs, Oil & Gas Journal 99 (27) (2001) 37–37.

[3] E. Brown, R. Thomas, A. Milne, The challenge of completing and stimulating horizontal wells, Oilfield Review 2 (3) (1990) 52–63.

[4] S. Bittleston, D. Guillot, Mud removal: research improves traditional cementing guidelines, Oilfield Review 3 (2) (1991) 44–54.

[5] I. A. Frigaard, K. G. Paso, P. R. de Souza Mendes, Bingham's model in the oil and gas industry, Rheologica Acta 56 (3) (2017) 259–282.

[6] M. Savery, R. Darbe, W. Chin, Modeling fluid interfaces during cementing using a 3D mud displacement simulator, in: Offshore Technology Conference, Offshore Technology Conference, 2007, pp. 1–7.

[7] S. Enayatpour, E. van Oort, Advanced modeling of cement displacement complexities, in: SPE/IADC Drilling Conference and Exhibition, Society of Petroleum Engineers, 2017, pp. 1–21.

[8] S. Taghavi, K. Alba, M. Moyers-Gonzalez, I. Frigaard, Incomplete fluid–fluid displacement of yield stress fluids in near-horizontal pipes: experiments and theory, Journal of Non-Newtonian Fluid Mechanics 167 (2012) 59–74.

[9] C.-W. Park, G. Homsy, Two-phase displacement in Hele Shaw cells: theory, Journal of Fluid Mechanics 139 (1984) 291–308.

[10] S. Bittleston, J. Ferguson, I. Frigaard, Mud removal and cement placement during primary cementing of an oil well–laminar non-Newtonian displacements in an eccentric annular Hele-Shaw cell, Journal of Engineering Mathematics 43 (2) (2002) 229–253.

[11] S. Pelipenko, I. Frigaard, Mud removal and cement placement during primary cementing of an oil well–part 2; steady-state displacements, Journal of Engineering Mathematics 48 (1) (2004) 1–26.

[12] S. Pelipenko, I. Frigaard, Visco-plastic fluid displacements in near-vertical narrow eccentric annuli: prediction of travelling-wave solutions and interfacial instability, Journal of Fluid Mechanics 520 (2004) 343–377.

[13] M. Escudier, P. Oliveira, F. Pinho, Fully developed laminar flow of purely viscous non-Newtonian liquids through annuli, including the effects of eccentricity and inner-cylinder rotation, International Journal of Heat and Fluid Flow 23 (1) (2002) 52–73.

[14] D. Jang, R. Jetli, S. Acharya, Comparison of the PISO, SIMPLER, and SIMPLEC algorithms for the treatment of the pressure-velocity coupling in steady flow problems, Numerical Heat Transfer, Part A: Applications 10 (3) (1986) 209–228.

[15] J. H. Edwards, V. G. R. Palla, K. B. Yerubandi, G. A. Fuller, M. R. Savery, et al., Prediction of residual cement in drill pipe after balanced plug job using finite difference 3D displacement simulator, in: SPE/IADC Middle East Drilling Technology Conference & Exhibition, Society of Petroleum Engineers, 2013, pp. 883–894.

[16] P. Tardy, N. Flamant, E. Lac, A. Parry, C. Sutama, S. Almagro, et al., New generation 3D simulator predicts realistic mud displacement in highly deviated and horizontal wells, in: SPE/IADC Drilling Conference and Exhibition, Society of Petroleum Engineers, 2017, pp. 1–18.

[17] W. Zhang, A. Almgren, V. Beckner, J. Bell, J. Blaschke, C. Chan, M. Day, B. Friesen, K. Gott, D. Graves, M. P. Katz, A. Myers, T. Nguyen, A. Nonaka, M. Rosso, S. Williams, M. Zingale, AMReX: a framework for block-structured adaptive mesh refinement, Journal of Open Source Software 4 (37) (2019) 1370.

[18] K. Sverdrup, N. Nikiforakis, A. Almgren, Highly parallelisable simulations of time-dependent viscoplastic fluid flow with structured adaptive mesh refinement, Physics of Fluids 30 (9) (2018) 093102.

[19] K. Sverdrup, N. Nikiforakis, A. Almgren, Numerical simulations of immiscible generalised Newtonian fluids, in: R. Owen, R. de Borst, J. Reese, C. Pearce (Eds.), Proceedings of the 6th European Conference on Computational Mechanics (ECCM VI) and 7th European Conference on Computational Fluid Dynamics (ECFD VII) 11–15 June 2018, Glasgow, UK, ECCOMAS, International Center for Numerical Methods in Engineering, 2019, pp. 2969–2980.

[20] K. Sverdrup, A. Almgren, N. Nikiforakis, An embedded boundary approach for efficient simulations of viscoplastic fluids in three dimensions, Physics of Fluids 31 (9) (2019) 093102.

[21] L. Bruno, L. Baker, Math and Mathematicians: The History of Math Discoveries Around the World, Vol. 4 of Math and Mathematicians Series, UXL, 2002.

[22] O. Reynolds, IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion, Philosophical Transactions of the Royal Society of London 186 (1895) 123–164.

[23] L. Euler, Principes généraux de l'état d'équilibre d'un fluide, Académie Royale des Sciences et des Belles Lettres de Berlin 11 (1757) 217–273.

[24] D. K. Clarke, H. Hassan, M. Salas, Euler calculations for multielement airfoils using Cartesian grids, AIAA journal 24 (3) (1986) 353–358.

[25] W. K. Anderson, J. L. Thomas, B. Van Leer, Comparison of finite volume flux vector splittings for the Euler equations, AIAA journal 24 (9) (1986) 1453–1460.

[26] G. P. Guruswamy, Unsteady aerodynamic and aeroelastic calculations for wings using Euler equations, AIAA journal 28 (3) (1990) 461–469.

[27] R. Laprise, The Euler equations of motion with hydrostatic pressure as an independent variable, Monthly Weather Review 120 (1) (1992) 197–207.

[28] N. Nikiforakis, J. Clarke, Numerical studies of the evolution of detonations, Mathematical and Computer Modelling 24 (8) (1996) 149–164.

[29] A. S. Almgren, J. B. Bell, P. Colella, T. Marthaler, A cartesian grid projection method for the incompressible Euler equations in complex geometries, SIAM Journal on Scientific Computing 18 (5) (1997) 1289–1309.

[30] D. Christodoulou, The Euler equations of compressible fluid flow, Bulletin of the American Mathematical Society 44 (4) (2007) 581–602.

[31] W. C. Skamarock, J. B. Klemp, A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, Journal of Computational Physics 227 (7) (2008) 3465–3485.

[32] M. M. Rai, Navier-Stokes simulations of rotor/stator interaction using patched and overlaid grids, Journal of Propulsion and Power 3 (5) (1987) 387–396.

[33] J. L. Thomas, W. K. Anderson, S. T. Krist, Navier-Stokes computations of vortical flows over low-aspect-ratio wings, AIAA journal 28 (2) (1990) 205–212.

[34] A. Jameson, L. Martinelli, N. Pierce, Optimum aerodynamic design using the Navier-Stokes equations, Theoretical and Computational Fluid Dynamics 10 (1-4) (1998) 213–237.

[35] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.

[36] M. Mihaescu, S. Murugappan, M. Kalra, S. Khosla, E. Gutmark, Large eddy simulation and Reynolds-averaged Navier-Stokes modeling of flow in a realistic pharyngeal airway model: An investigation of obstructive sleep apnea, Journal of Biomechanics 41 (10) (2008) 2279–2288.

[37] J. Deiber, W. Schowalter, Flow through tubes with sinusoidal axial variations in diameter, AIChE Journal 25 (4) (1979) 638–645.

[38] G. Vinay, A. Wachs, J.-F. Agassant, Numerical simulation of weakly compressible Bingham flows: the restart of pipeline flows of waxy crude oils, Journal of Non-Newtonian Fluid Mechanics 136 (2) (2006) 93–105.

[39] M. B. Cardenas, D. T. Slottke, R. A. Ketcham, J. M. Sharp, Navier-Stokes flow and transport simulations using real fractures shows heavy tailing due to eddies, Geophysical Research Letters 34 (14).

[40] F. Boyer, C. Lapuerta, S. Minjeaud, B. Piar, M. Quintard, Cahn–Hilliard/Navier–Stokes model for the simulation of three-phase flows, Transport in Porous Media 82 (3) (2010) 463–483.

[41] C. L. M. H. Navier, Memoire sur les lois du mouvement des fluides, Mémoires de l'Académie Royale des Sciences de l'Institut de France 6 (1822) 389–440.

[42] G. G. Stokes, On the theories of the internal friction of fluids in motion and of the equilibrium and motion of elastic solids, Transactions of the Cambridge Philosophical Society 8 (1845) 287–319.

[43] C. L. Fefferman, Existence and smoothness of the Navier-Stokes equation, The Millennium Prize Problems (2006) 57–67.

[44] G. Emanuel, Bulk viscosity in the Navier–Stokes equations, International Journal of Engineering Science 36 (11) (1998) 1313–1323.

[45] H. A. Barnes, J. F. Hutton, K. Walters, An introduction to rheology, Vol. 3, Elsevier, 1989.

[46] J. Mewis, Thixotropy—a general review, Journal of Non-Newtonian Fluid Mechanics 6 (1) (1979) 1–20.

[47] H. A. Barnes, Thixotropy—a review, Journal of Non-Newtonian Fluid Mechanics 70 (1-2) (1997) 1–33.

[48] S. S. Pegler, N. J. Balmforth, Locomotion over a viscoplastic film, Journal of Fluid Mechanics 727 (2013) 1–29.

[49] M. W. Denny, A quantitative model for the adhesive locomotion of the terrestrial slug, ariolimax columbianus, Journal of experimental Biology 91 (1) (1981) 195–217.

[50] A. J. Apostolidis, A. N. Beris, Modeling of the blood rheology in steady-state shear flows, Journal of Rheology 58 (3) (2014) 607–633.

[51] A. J. Apostolidis, M. J. Armstrong, A. N. Beris, Modeling of human blood rheology in transient shear flows, Journal of Rheology 59 (1) (2015) 275–298.

[52] A. J. Apostolidis, A. N. Beris, The effect of cholesterol and triglycerides on the steady state shear rheology of blood, Rheologica Acta 55 (6) (2016) 497–509.

[53] A. J. Apostolidis, A. P. Moyer, A. N. Beris, Non-Newtonian effects in simulations of coronary arterial blood flow, Journal of Non-Newtonian Fluid Mechanics 233 (2016) 155–165.

[54] T. Schwedoff, La rigidité des fluides, Rapports du Congrès intern. de Physique 1 (1900) 478–486.

[55] E. C. Bingham, An investigation of the laws of plastic flow, Bulletin of the Bureau of Standards 13 (2) (1916) 309–353.

[56] J. Oldroyd, A rational formulation of the equations of plastic flow for a Bingham solid, in: Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 43, Cambridge University Press, 1947, pp. 100–105.

[57] W. H. Herschel, R. Bulkley, Konsistenzmessungen von Gummi-Benzollösungen, Colloid & Polymer Science 39 (4) (1926) 291–300.

[58] R. Christensen, Theory of viscoelasticity: an introduction, Elsevier, 2012.

[59] W. Casson, A flow equation for pigment-oil suspensions of printing of the printing ink type. Rheology of Dispersed Systems (1959).

[60] P. de Souza Mendes, E. S. Dutra, Viscosity function for yield-stress liquids, Applied Rheology 14 (6) (2004) 296–302.

[61] M. Dinkgreve, J. Paredes, M. M. Denn, D. Bonn, On different ways of measuring “the” yield stress, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 233–241.

[62] H. A. Barnes, The yield stress – a review or ‘$\pi\alpha\nu\tau\alpha\ \rho\varepsilon\iota$’ – everything flows?, Journal of Non-Newtonian Fluid Mechanics 81 (1) (1999) 133–178.

[63] N. J. Balmforth, I. A. Frigaard, G. Ovarlez, Yielding to stress: recent developments in viscoplastic fluid mechanics, Annual Review of Fluid Mechanics 46 (2014) 121–146.

[64] A. Syrakos, G. C. Georgiou, A. N. Alexandrou, Cessation of the lid-driven cavity flow of Newtonian and Bingham fluids, Rheologica Acta 55 (1) (2016) 51–66.

[65] P. Moller, A. Fall, V. Chikkadi, D. Derks, D. Bonn, An attempt to categorize yield stress fluid behaviour, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 367 (1909) (2009) 5139–5155.

[66] H. S. Tang, D. M. Kalyon, Estimation of the parameters of Herschel-Bulkley fluid under wall slip using a combination of capillary and squeeze flow viscometers, Rheologica Acta 43 (1) (2004) 80–88.

[67] L. Muravleva, Axisymmetric squeeze flow of a viscoplastic Bingham medium, Journal of Non-Newtonian Fluid Mechanics 249 (2017) 97–120.

[68] L. Muravleva, Squeeze flow of Bingham plastic with stick-slip at the wall, Physics of Fluids 30 (3) (2018) 030709.

[69] O. Reynolds, XXIX. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels, Philosophical Transactions of the Royal society of London 174 (1883) 935–982.

[70] I. Frigaard, S. Howison, I. Sobey, On the stability of Poiseuille flow of a Bingham fluid, Journal of Fluid Mechanics 263 (1991) 133–150.

[71] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, in: Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 43, Cambridge University Press, 1947, pp. 50–67.

[72] C. Runge, Über die numerische Auflösung von Differentialgleichungen, Mathematische Annalen 46 (2) (1895) 167–178.

[73] W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, Zeitschrift für Mathematik und Physik 46 (1901) 435–453.

[74] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.

[75] P. Colella, P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, Journal of Computational Physics 54 (1) (1984) 174–201.

[76] J.-L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering 195 (44) (2006) 6011–6045.

[77] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Mathematics of Computation 22 (104) (1968) 745–762.

[78] N. N. Yanenko, The method of fractional steps, Springer, 1971.

[79] E. F. Toro, V. A. Titarev, Derivative Riemann solvers for systems of conservation laws and ADER methods, Journal of Computational Physics 212 (1) (2006) 150–165.

[80] D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 168 (2) (2001) 464–499.

[81] E. Weinan, J.-G. Liu, et al., Gauge method for viscous incompressible flows, Communications in Mathematical Sciences 1 (2) (2003) 317–332.

[82] J. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, Journal of Computational Physics 192 (1) (2003) 262–276.

[83] H. Helmholtz, Über Integrale der hydrodynamischen Gleichungen, welche den Wirbelbewegungen entsprechen, Journal für die reine und angewandte Mathematik 55 (1858) 25–55.

[84] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass transfer 15 (10) (1972) 1787–1806.

[85] M.-J. Ni, M. A. Abdou, A bridge between projection methods and SIMPLE type methods for incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering 72 (12) (2007) 1490–1512.

[86] O. Zikanov, Essential Computational Fluid Dynamics, John Wiley & Sons, 2010.

[87] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, 1980.

[88] J. Van Doormaal, G. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, Numerical Heat Transfer 7 (2) (1984) 147–163.

[89] P. Saramito, A. Wachs, Progress in numerical simulation of yield stress fluid flows, Rheologica Acta 56 (3) (2017) 211–230.

[90] M. Bercovier, M. Engelman, A finite-element method for incompressible non-Newtonian flows, Journal of Computational Physics 36 (3) (1980) 313–326.

[91] R. Tanner, J. Milthorpe, Numerical simulation of the flow of fluids with yield stress, in: C. Taylor, C. Johnson, I. Smith (Eds.), Proceedings of Third International Conference on Numerical Methods in Laminar and Turbulent Flow, Pineridge Press, Swansea, UK, 1983, pp. 680–690.

[92] T. C. Papanastasiou, Flows of materials with yield, Journal of Rheology 31 (5) (1987) 385–404.

[93] E. Mitsoulis, T. Zisis, Flow of Bingham plastics in a lid-driven square cavity, Journal of Non-Newtonian Fluid Mechanics 101 (1) (2001) 173–180.

[94] T. Zisis, E. Mitsoulis, Viscoplastic flow around a cylinder kept between parallel plates, Journal of Non-Newtonian Fluid Mechanics 105 (1) (2002) 1–20.

[95] E. Mitsoulis, On creeping drag flow of a viscoplastic fluid past a circular cylinder: wall effects, Chemical Engineering Science 59 (4) (2004) 789–800.

[96] E. Mitsoulis, J. Tsamopoulos, Numerical simulations of complex yield-stress fluid flows, Rheologica Acta 56 (3) (2017) 231–258.

[97] A. Syrakos, G. C. Georgiou, A. N. Alexandrou, Solution of the square lid-driven cavity flow of a Bingham plastic using the finite volume method, Journal of Non-Newtonian Fluid Mechanics 195 (2013) 19–31.

[98] A. Syrakos, G. C. Georgiou, A. N. Alexandrou, Performance of the finite volume method in solving regularised Bingham flows: Inertia effects in the lid-driven cavity flow, Journal of Non-Newtonian Fluid Mechanics 208 (2014) 88–107.

[99] M. Fortin, R. Glowinski, Chapter III: On decomposition-coordination methods using an augmented Lagrangian, in: M. Fortin, R. Glowinski (Eds.), Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, Vol. 15 of Studies in Mathematics and Its Applications, Elsevier, 1983, pp. 97 – 146.

[100] G. Duvaut, J. L. Lions, Inequalities in mechanics and physics, Vol. 219, Springer, 1976.

[101] M. R. Hestenes, Multiplier and gradient methods, Journal of Optimization Theory and Applications 4 (5) (1969) 303–320.

[102] P. Saramito, N. Roquet, An adaptive finite element method for viscoplastic fluid flows in pipes, Computer Methods in Applied Mechanics and Engineering 190 (40) (2001) 5391–5412.

[103] D. Vola, L. Boscardin, J. Latché, Laminar unsteady flows of Bingham fluids: a numerical strategy and some benchmark results, Journal of Computational Physics 187 (2) (2003) 441–456.

[104] T. Treskatis, M. A. Moyers-Gonzalez, C. J. Price, An accelerated dual proximal gradient method for applications in viscoplasticity, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 115–130.

[105] P. Saramito, A damped Newton algorithm for computing viscoplastic fluid flows, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 6–15.

[106] J. Bleyer, Advances in the simulation of viscoplastic fluid flows using interior-point methods, Computer Methods in Applied Mechanics and Engineering 330 (2018) 368–394.

[107] Y. Dimakopoulos, G. Makrigiorgos, G. Georgiou, J. Tsamopoulos, The PAL (Penalized Augmented Lagrangian) method for computing viscoplastic flows: a new fast converging scheme, Journal of Non-Newtonian Fluid Mechanics.

[108] H. Barnes, K. Walters, The yield stress myth?, Rheologica Acta 24 (4) (1985) 323–326.

[109] J. P. Hartnett, R. Y. Hu, The yield stress – an engineering reality, Journal of Rheology 33 (4) (1989) 671–679.

[110] M. Dinkgreve, M. M. Denn, D. Bonn, "everything flows?”: elastic effects on startup flows of yield-stress fluids, Rheologica Acta 56 (3) (2017) 189–194.

[111] I. Frigaard, C. Nouar, On the usage of viscosity regularisation methods for visco-plastic fluid flow computation, Journal of Non-Newtonian Fluid Mechanics 127 (1) (2005) 1–26.

[112] T. Treskatis, Fast proximal algorithms for applications in viscoplasticity., Ph.D. thesis, University of Canterbury (2016).

[113] H. H. Hu, Computational fluid dynamics, in: Fluid Mechanics, Elsevier, 2012, pp. 421–472.

[114] T. Stolarski, Y. Nakasone, S. Yoshimoto, Engineering analysis with ANSYS software, Butterworth-Heinemann, 2018.

[115] COMSOL, Introduction to COMSOL Multiphysics®, Tech. rep., COMSOL Multiphysics, Burlington, MA (1998).

[116] H. Jasak, A. Jemcov, Z. Tukovic, et al., OpenFOAM: A C++ library for complex physics simulations, in: International workshop on coupled methods in numerical dynamics, Vol. 1000, IUC Dubrovnik, Croatia, 2007, pp. 1–20.

[117] A. Logg, K.-A. Mardal, G. Wells, Automated solution of differential equations by the finite element method: the FEniCS book, Vol. 84, Springer Science & Business Media, 2012.

[118] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The FEniCS project version 1.5, Archive of Numerical Software 3 (100) (2015) 9–23.

[119] K. Moreland, R. Oldfield, Formal metrics for large-scale parallel performance, in: International Conference on High Performance Computing, Springer, 2015, pp. 488–496.

[120] J. Dongarra, S. Gottlieb, W. T. Kramer, Race to exascale, Computing in Science & Engineering 21 (1) (2019) 4–5.

[121] A. Almgren, V. Beckner, J. Bell, M. Day, L. Howell, C. Joggerst, M. Lijewski, A. Nonaka, M. Singer, M. Zingale, CASTRO: A new compressible astrophysical solver. I. Hydrodynamics and self-gravity, The Astrophysical Journal 715 (2) (2010) 1221.

[122] A. S. Almgren, J. B. Bell, M. J. Lijewski, Z. Lukić, E. Van Andel, Nyx: A massively parallel AMR code for computational cosmology, The Astrophysical Journal 765 (1) (2013) 39.

[123] A. Dubey, A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, et al., A survey of high level frameworks in block-structured adaptive mesh refinement packages, Journal of Parallel and Distributed Computing 74 (12) (2014) 3217–3227.

[124] S. Habib, A. Pope, H. Finkel, N. Frontiere, K. Heitmann, D. Daniel, P. Fasel, V. Morozov, G. Zagaris, T. Peterka, V. Vishwanath, Z. Lukić, S. Sehrish, W.-k. Liao, HACC: Simulating sky surveys on state-of-the-art supercomputing architectures, New Astronomy 42 (2016) 49–65.

[125] P. Wesseling, C. W. Oosterlee, Geometric multigrid with applications to computational fluid dynamics, Journal of Computational and Applied Mathematics 128 (1) (2001) 311–334.

[126] W. L. Briggs, S. F. McCormick, et al., A multigrid tutorial, Vol. 72, SIAM, 2000.

[127] H. A. Van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (2) (1992) 631–644.

[128] J. Ahrens, B. Geveci, C. Law, Paraview: An end-user tool for large data visualization, The Visualization Handbook 717.

[129] M. J. Turk, B. D. Smith, J. S. Oishi, S. Skory, S. W. Skillman, T. Abel, M. L. Norman, yt: A multi-code analysis toolkit for astrophysical simulation data, The Astrophysical Journal Supplement Series 192 (1) (2010) 9.

[130] H. Childs, VisIt: An end-user tool for visualizing and analyzing very large data, Lawrence Berkeley National Laboratory (2012).

[131] S. Abhyankar, J. Brown, E. M. Constantinescu, D. Ghosh, B. F. Smith, H. Zhang, PETSc/TS: A modern scalable ODE/DAE solver library, arXiv preprint arXiv:1806.01437.

[132] R. D. Falgout, U. M. Yang, hypre: A library of high performance preconditioners, in: International Conference on Computational Science, Springer, 2002, pp. 632–641.

[133] J.-L. Vay, A. Almgren, J. Bell, L. Ge, D. Grote, M. Hogan, O. Kononenko, R. Lehe, A. Myers, C. Ng, et al., Warp-X: A new exascale computing platform for beam–plasma simulations, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 909 (2018) 476–479.

[134] M. Zingale, A. Almgren, M. B. Sazo, V. Beckner, J. Bell, B. Friesen, A. Jacobs, M. Katz, C. Malone, A. Nonaka, et al., Meeting the challenges of modeling astrophysical thermonuclear explosions: Castro, Maestro, and the AMReX astrophysics suite, in: Journal of Physics: Conference Series, Vol. 1031, IOP Publishing, 2018, p. 012024.

[135] M. Day, S. Tachibana, J. Bell, M. Lijewski, V. Beckner, R. K. Cheng, A combined computational and experimental characterization of lean premixed turbulent low swirl laboratory flames: I. methane flames, Combustion and Flame 159 (1) (2012) 275–290.

[136] M. Day, S. Tachibana, J. Bell, M. Lijewski, V. Beckner, R. K. Cheng, A combined computational and experimental characterization of lean premixed turbulent low swirl laboratory flames II. hydrogen flames, Combustion and Flame 162 (5) (2015) 2148–2165.

[137] M. Syamlal, W. Rogers, T. J. OBrien, MFIX documentation theory guide, Tech. rep., USDOE Morgantown Energy Technology Center, WV (United States) (1993).

[138] A. Almgren, J. Bell, P. Colella, L. Howell, M. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, Journal of Computational Physics 142 (1998) 1–46.

[139] M. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Transactions on Systems, Man, and Cybernetics 21 (5) (1991) 1278–1286.

[140] R. Pember, L. Howell, J. Bell, P. Colella, W. Crutchfield, W. Fiveland, J. Jessee, An adaptive projection method for unsteady, low-Mach number combustion, Combustion Science and Technology 140 (1-6) (1998) 123–168.

[141] M. Day, J. Bell, Numerical simulation of laminar reacting flows with complex chemistry, Combustion Theory and Modelling 4 (4) (2000) 535–556.

[142] M. Lai, J. Bell, P. Colella, A projection method for combustion in the zero Mach number limit, in: 11th Computational Fluid Dynamics Conference, 1993, p. 3369.

[143] A. S. Almgren, J. B. Bell, W. G. Szymczak, A numerical method for the incompressible Navier-Stokes equations based on an approximate projection, SIAM Journal on Scientific Computing 17 (2) (1996) 358–369.

[144] D. F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, Journal of Computational Physics 163 (2) (2000) 271–312.

[145] P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, SIAM Journal on Scientific and Statistical Computing 6 (1) (1985) 104–117.

[146] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, Mathematische Annalen 100 (1) (1928) 32–74.

[147] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[148] B. Lalanne, L. R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, Journal of Computational Physics 301 (2015) 289–307.

[149] M. Kang, R. P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, Journal of Scientific Computing 15 (3) (2000) 323–360.

[150] J. Howard, An Appraisal of Rayleigh, Vol. 11, Air Force Cambridge Research Laboratories, Office of Aerospace Research, United States Air Force, 1964.

[151] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, Journal of Computational Physics 85 (2) (1989) 257–283.

[152] M. Dumbser, I. Peshkov, E. Romenski, O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat-conducting fluids and elastic solids, Journal of Computational Physics 314 (2016) 824–862.

[153] A. Wachs, Computational methods for viscoplastic fluid flows, in: Lectures on Visco-Plastic Fluid Mechanics, Springer, 2019, pp. 83–125.

[154] E. Muravleva, M. Olshanskii, Two finite-difference schemes for calculation of Bingham fluid flows in a cavity, Russian Journal of Numerical Analysis and Mathematical Modelling 23 (6) (2008) 615–634.

[155] M. A. Olshanskii, Analysis of semi-staggered finite-difference method with application to Bingham flows, Computer Methods in Applied Mechanics and Engineering 198 (9-12) (2009) 975–985.

[156] K. D. Housiadas, Improved convergence based on linear and non-linear transformations at low and high Weissenberg asymptotic analysis, Journal of Non-Newtonian Fluid Mechanics 247 (2017) 1–14.

[157] B. C. Bell, K. S. Surana, p-version least squares finite element formulation for two-dimensional, incompressible, non-Newtonian isothermal and non-isothermal fluid flow, International Journal for Numerical Methods in Fluids 18 (2) (1994) 127–162.

[158] P. Neofytou, A 3rd order upwind finite volume method for generalised Newtonian fluid flows, Advances in Engineering Software 36 (10) (2005) 664–680.

[159] U. Ghia, K. N. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, Journal of Computational Physics 48 (3) (1982) 387–411.

[160] Z. Chai, B. Shi, Z. Guo, F. Rong, Multiple-relaxation-time lattice Boltzmann model for generalized Newtonian fluid flows, Journal of Non-Newtonian Fluid Mechanics 166 (5) (2011) 332–342.

[161] P. R. de Souza Mendes, Dimensionless non-Newtonian fluid mechanics, Journal of Non-Newtonian Fluid Mechanics 147 (1-2) (2007) 109–116.

[162] R. L. Thompson, E. J. Soares, Viscoplastic dimensionless numbers, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 57–64.

[163] E. J. Dean, R. Glowinski, G. Guidoboni, On the numerical simulation of Bingham visco-plastic flow: old and new results, Journal of Non-Newtonian Fluid Mechanics 142 (1-3) (2007) 36–62.

[164] L. Muravleva, Uzawa-like methods for numerical modeling of unsteady viscoplastic Bingham medium flows, Applied Numerical Mathematics 93 (2015) 140–149.

[165] Y. Dimakopoulos, J. Tsamopoulos, Transient displacement of a viscoplastic material by air in straight and suddenly constricted tubes, Journal of Non-Newtonian Fluid Mechanics 112 (1) (2003) 43–75.

[166] R. N. Elias, M. A. Martins, A. L. Coutinho, Parallel edge-based solution of viscoplastic flows with the SUPG/PSPG formulation, Computational Mechanics 38 (4-5) (2006) 365–381.

[167] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.

[168] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational Physics 169 (2) (2001) 463–502.

[169] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of Computational Physics 152 (2) (1999) 457–492.

[170] T. Liu, B. Khoo, K. Yeo, Ghost fluid method for strong shock impacting on material interface, Journal of Computational Physics 190 (2) (2003) 651–681.

[171] C. Wang, T. Liu, B. Khoo, A real ghost fluid method for the simulation of multimedium compressible flow, SIAM Journal on Scientific Computing 28 (1) (2006) 278–302.

[172] S. K. Sambasivan, H. Udaykumar, Ghost fluid method for strong shock interactions part 1: Fluid-fluid interfaces, AIAA Journal 47 (12) (2009) 2907–2922.

[173] E. S. Oran, J. P. Boris, Numerical simulation of reactive flow, Cambridge University Press, 2005.

[174] R. J. Kee, M. E. Coltrin, P. Glarborg, Chemically reacting flow: theory and practice, John Wiley & Sons, 2005.

[175] W. F. Noh, P. Woodward, SLIC (simple line interface calculation), in: Proceedings of the fifth international conference on numerical methods in fluid dynamics June 28–July 2, 1976 Twente University, Enschede, Netherlands, Springer, 1976, pp. 330–340.

[176] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1) (1981) 201–225.

[177] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1) (1999) 81–124.

[178] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2) (2000) 301–337.

[179] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet, International Journal of Multiphase Flow 33 (5) (2007) 510–524.

[180] D. H. Sharp, An overview of Rayleigh-Taylor instability, Tech. rep., Los Alamos National Lab., NM (USA) (1983).

[181] G. Tryggvason, Numerical simulations of the Rayleigh-Taylor instability, Journal of Computational Physics 75 (2) (1988) 253–282.

[182] J.-L. Guermond, L. Quartapelle, A projection FEM for variable density incompressible flows, Journal of Computational Physics 165 (1) (2000) 167–188.

[183] B. S. Hosseini, S. Turek, M. Möller, C. Palmes, Isogeometric analysis of the Navier–Stokes–Cahn–Hilliard equations with application to incompressible two-phase flows, Journal of Computational Physics 348 (2017) 171–194.

[184] K. Wielage-Burchard, I. Frigaard, Static wall layers in plane channel displacement flows, Journal of Non-Newtonian Fluid Mechanics 166 (5-6) (2011) 245–261.

[185] A. Koblitz, S. Lovett, N. Nikiforakis, Direct numerical simulation of particle sedimentation in a Bingham fluid, Physical Review Fluids 3 (9) (2018) 093302.

[186] N. Kanaris, S. Kassinos, A. Alexandrou, On the transition to turbulence of a viscoplastic fluid past a confined cylinder: A numerical study, International Journal of Heat and Fluid Flow 55 (2015) 65–75.

[187] E. Chaparian, A. Wachs, I. A. Frigaard, Inline motion and hydrodynamic interaction of 2d particles in a viscoplastic fluid, Physics of Fluids 30 (3) (2018) 033101.

[188] A. Roosing, O. Strickson, N. Nikiforakis, Fast distance fields for fluid dynamics mesh generation on graphics hardware, arXiv preprint arXiv:1903.00353.

[189] J. W. Purvis, J. E. Burkhalter, Prediction of critical Mach number for store configurations, AIAA Journal 17 (11) (1979) 1170–1177.

[190] B. Wedan, J. South, Jr., A method for solving the transonic full-potential equation for general configurations, in: 6th Computational Fluid Dynamics Conference Danvers, 1983, p. 1889.

[191] S. Falle, J. Giddings, An adaptive multigrid applied to supersonic blunt body flow, Numerical Methods in Fluid Dynamics, III.

[192] M. Berger, R. Leveque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, in: 9th Computational Fluid Dynamics Conference, 1989, p. 1930.

[193] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, Journal of Computational Physics 211 (1) (2006) 347–366.

[194] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1) (1998) 60–85.

[195] P. McCorquodale, P. Colella, H. Johansen, A Cartesian grid embedded boundary method for the heat equation on irregular domains, Journal of Computational Physics 173 (2) (2001) 620–635.

[196] M. Berger, Cut cells: meshes and solvers, in: Handbook of Numerical Analysis, Vol. 18, Elsevier, 2017, pp. 1–22.

[197] N. Gokhale, N. Nikiforakis, R. Klein, A dimensionally split Cartesian cut cell method for hyperbolic conservation laws, Journal of Computational Physics 364 (2018) 186–208.

[198] N. Gokhale, N. Nikiforakis, R. Klein, A dimensionally split Cartesian cut cell method for the compressible Navier–Stokes equations, Journal of Computational Physics 375 (2018) 1205–1219.

[199] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of Fluids 8 (12) (1965) 2182–2189.

[200] K. Balakrishnan, A. L. Garcia, A. Donev, J. B. Bell, Fluctuating hydrodynamics of multispecies nonreactive mixtures, Physical Review E 89 (1) (2014) 013017.

[201] A. S. Almgren, J. B. Bell, W. Y. Crutchfield, Approximate projection methods: Part I. inviscid analysis, SIAM Journal on Scientific Computing 22 (4) (2000) 1139–1159.

[202] D. Graves, P. Colella, D. Modiano, J. Johnson, B. Sjogreen, X. Gao, A Cartesian grid embedded boundary method for the compressible Navier–Stokes equations, Communications in Applied Mathematics and Computational Science 8 (1) (2013) 99–122.

[203] R. J. LeVeque, High resolution finite volume methods on arbitrary grids via wave propagation, Journal of Computational Physics 78 (1) (1988) 36–63.

[204] R. J. LeVeque, Cartesian grid methods for flow in irregular regions, Numerical Methods in Fluid Dynamics 3 (1988) 375–382.

[205] R. W. Ansley, T. N. Smith, Motion of spherical particles in a Bingham plastic, AIChE Journal 13 (6) (1967) 1193–1196.

[206] A. Beris, J. Tsamopoulos, R. Armstrong, R. Brown, Creeping motion of a sphere through a Bingham plastic, Journal of Fluid Mechanics 158 (1985) 219–244.

[207] J. Blackery, E. Mitsoulis, Creeping motion of a sphere in tubes filled with a Bingham plastic material, Journal of Non-Newtonian Fluid Mechanics 70 (1-2) (1997) 59–77.

[208] B. T. Liu, S. J. Muller, M. M. Denn, Convergence of a regularization method for creeping flow of a Bingham material about a rigid sphere, Journal of Non-Newtonian Fluid Mechanics 102 (2) (2002) 179–191.

[209] D. L. Tokpavi, A. Magnin, P. Jay, Very slow flow of Bingham viscoplastic fluid around a circular cylinder, Journal of Non-Newtonian Fluid Mechanics 154 (1) (2008) 65–76.

[210] E. Chaparian, I. A. Frigaard, Yield limit analysis of particle motion in a yield-stress fluid, Journal of Fluid Mechanics 819 (2017) 311–351.

[211] A. R. Koblitz, S. Lovett, N. Nikiforakis, Viscoplastic squeeze flow between two identical infinite circular cylinders, Physical Review Fluids 3 (2) (2018) 023301.

[212] Prashant, J. Derksen, Direct simulations of spherical particle motion in Bingham liquids, Computers & Chemical Engineering 35 (7) (2011) 1200–1214.

[213] S.-G. Chen, C.-H. Zhang, Y.-T. Feng, Q.-C. Sun, F. Jin, Three-dimensional simulations of Bingham plastic flows with the multiple-relaxation-time lattice Boltzmann model, Engineering Applications of Computational Fluid Mechanics 10 (1) (2016) 346–358.

[214] K. Adachi, N. Yoshioka, On creeping flow of a visco-plastic fluid past a circular cylinder, Chemical Engineering Science 28 (1) (1973) 215–226.

[215] M. F. Randolph, G. Houlsby, The limiting pressure on a circular pile loaded laterally in cohesive soil, Geotechnique 34 (4) (1984) 613–623.

[216] B. T. Liu, S. J. Muller, M. M. Denn, Interactions of two rigid spheres translating collinearly in creeping flow in a Bingham material, Journal of Non-Newtonian Fluid Mechanics 113 (1) (2003) 49–67.

[217] H. Faxén, Forces exerted on a rigid cylinder in a viscous fluid between two parallel fixed planes, Ingeniörsvetenskapsakademiens Handlingar 187 (1946) 1–13.

[218] P. Davidson, Turbulence: an introduction for scientists and engineers, Oxford University Press, 2015.

[219] S. De, J. Kuipers, E. Peters, J. Padding, Viscoelastic flow simulations in model porous media, Physical Review Fluids 2 (5) (2017) 053303.

[220] J. Oldroyd, Two-dimensional plastic flow of a Bingham solid: a plastic boundary-layer theory for slow motion, in: Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 43, Cambridge University Press, 1947, pp. 383–395.

[221] N. J. Balmforth, Viscoplastic asymptotics and other analytical methods, in: Lectures on Visco-Plastic Fluid Mechanics, Springer, 2019, pp. 41–82.

[222] G. G. Stokes, On the effect of the internal friction of fluids on the motion of pendulums, Transactions of the Cambridge Philosophical Society 9 (1851) 8.