Mixed Viscosity Methods for Accelerated Viscous Fluid Simulation

by

Daniel Winters

A research project
presented to the University of Waterloo
in fulfillment of the
research paper requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners.

I understand that my report may be made electronically available to the public.

Abstract

Fluid simulation has increasingly important applications in computer graphics, animation, gaming, and virtual reality. As our virtual and animated worlds become more complex, the ability to model and animate viscous fluid flow in an efficient manner is crucial. The main hallmarks of a successful fluid simulation are computational efficiency and correct realistic behaviour. For simulating highly viscous fluids such as mud, tar, lava, molasses, and honey, the inclusion of the viscosity component of the incompressible Navier-Stokes equations can decrease the efficiency of the fluid solver by several orders of magnitude. Stable implicit methods to deal with the viscosity portion of fluid solvers have been developed since the early 2000's and are incorporated into standard animation software, yet computational efficiency continues to be a limiting factor in the use of these methods.

The primary objective of this study is to develop an efficient "drop-in" replacement of the operator-split viscosity solve of the incompressible Navier-Stokes fluid equations. We present an approach that exploits efficiency by using a cheaper discretization in the interior of the liquid, and preserves realistic viscous fluid behaviours by employing a more expensive but accurate discretization near the air-liquid interface.

We propose and evaluate three novel mixed discretization approaches to do so, resulting in both SPD and non-symmetric matrices. We evaluate all three mixed discretization's ability to solve the two PDEs in their respective regions via a mixed spatial convergence test of the viscosity step alone. We see that the non-symmetric mixed formulation passes six of six convergence tests while the two SPD formulations only pass one of the six. We subsequently integrate two of these hybrid viscosity discretizations into a comprehensive 2D fluid solver. We then assess both the computational expenses and the associated qualitative behaviors of the mixed viscosity models, comparing them against the benchmark models that do not incorporate mixing. Compared to the more expensive benchmark viscosity model, we see that these mixed methods always incur computational cost benefits for a given fixed solver, and that overall cost benefits can occur while qualitatively retaining viscous behaviours.

Acknowledgements

I would like to extend my sincere gratitude to my mentor, Professor Christopher Batty, for his guidance and invaluable support throughout this project.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fluid simulation in the context of computer graphics plays a vital role in films, gaming, and virtual reality through the use of computer graphics software. The key criteria for a successful fluid simulation lie in achieving both computational efficiency and accurate realistic behavior. However, when dealing with highly viscous fluids and incorporating the viscosity component into the incompressible Navier-Stokes equations, the efficiency of a given fluid solver often experiences a significant decline from solving a linear system three times the size of the pressure projection in 3D. This inclusion of the viscosity component is necessary for modeling viscous fluids like mud, lava, tar, molasses, honey, and many more. Stable grid based implicit methods for addressing the viscosity aspect have been under development since the early 2000s, integrated into standard animation software [25]. Nevertheless, efficiency remains the primary limiting factor for these methods and we aim to address this through the use of a mixed discretization scheme.

The primary objective of this study is to develop an efficient drop-in replacement of the operator-split viscosity solve of the Navier-Stokes fluid equations to (1) exploit efficiency in the interior of the fluid by using the less accurate yet more efficient discretization of Carlson et al. [7], and (2) employ the more accurate discretization of Batty and Bridson [4] near the surface of the fluid to achieve the correct boundary condition to preserve known viscous fluid behaviours. After covering related work and mathematical background in Section 2, we propose and evaluate three novel mixed discretization coupling schemes to do so in Section 3. Two of the three mixed methods yield SPD matrices. We develop a novel *mixed* discretization spatial convergence test in Section 4.1 of the altered viscosity steps to test each mixed method's ability to solve the two PDEs in their respective regions. In Section 4.2 we then integrate two of these hybrid viscosity methods, one SPD and one non-symmetric method, into a comprehensive 2D fluid solver. We then assess both the computational expenses and the associated qualitative behaviors of the mixed viscosity models, comparing them against benchmarks that do not incorporate mixing.

Chapter 2

Background

This chapter covers the necessary background information of this research project. Related work in Eulerian viscosity and efficiency acceleration techniques for both viscosity and pressure projection are presented. We then present the Mathematical background necessary for this project. We briefly discuss general methods and techniques for full fluid solvers. Next, an in-depth mathematical background on Eulerian viscosity integration is covered, which is what this project aims to accelerate the efficiency of through a mixed treatment of the presented models.

2.1 Related Work

Eulerian Viscous Animation in Computer Graphics. Harlow and Welch in 1965 began the initial attempts at modelling viscous fluids from solving the Navier-Stokes fluid equations in the context of Computational Fluid Dynamics (CFD) [14]. Miller and Pearce [18] introduced viscous fluids to the field of computer graphics by approximating melting and flowing viscous substances via extending particle systems with inter-particle spring forces. Similarly, Terzopoulos et al. [26] showcased a finite element method which melted solids into molten-state fluids with long-range attraction and short-range repulsive forces between particles. These previous two techniques used ad hoc methods to achieve viscous effects which were not derived by tackling the viscosity portion of a Navier-Stokes solver. Building upon Harlow and Welch's work on MAC grid fluid simulation, Foster and Metaxas [11] were the first to use the 3D Navier-Stokes equations to simulate viscous fluids, with explicit time integration requiring small time steps. The first implicit time integration scheme for viscosity was introduced by Stam [15], in which they made the simplification that viscosity is constant throughout the fluid to reduce the viscosity equation to three separate uncoupled diffusion-like symmetric positive definite (SPD) linear systems for each velocity component rather than one much larger system of coupled velocity components. This was extended to be used with level sets to accurately track the liquid surface by Foster and Fedkiw [10]. Carlson et al. further extended this diffusion-like viscosity treatment with free surfaces as well as a temperature-dependent variable viscosity model throughout the liquid [7]. In the results there existed artificial dampening of liquid surrounded by air, which was manually rectified. Fält and Roble resolved the artificial dampening from not neglecting solid boundary contributions to the linear system matrix diagonal [9].

The decoupled implicit velocity scheme in Carlson et al. [7] which arises from assuming constant viscosity will be referred to as ‘Laplacian Viscosity’ or ‘Laplacian Form Viscosity’ throughout this project since the three separate SPD systems each arise from the 3D Laplacian operator acting on the velocity field. Unfortunately the free surface boundary condition which arises in Laplacian Viscosity fails to yield correct viscous fluid behaviours such as buckling, rotating and bending. This is due to the decoupled nature of the Laplacian Viscosity which enforces the incorrect free surface boundary condition. Batty and Bridson fix this by not assuming constant viscosity to reduce the shear stress tensor, leaving the velocity components coupled to one another in the viscosity solve to give one much larger SPD system involving all velocity components simultaneously [4]. This was the first implicit treatment of viscosity which did not drop the velocity-coupling cross term, and we refer to this velocity-coupled form as the ‘Full Form Viscosity’ throughout this project. Batty and Bridson use a variational principle to automatically enforce the zero-traction free surface condition at the liquid-air boundary for the operator-split viscosity equation and they show the equivalence between this discrete minimization approach with the time-discretized operator-split viscosity equation [4]. The correct enforcement of the operator-split version of the zero-traction free surface boundary condition in [4] is what yields visually compelling buckling, rotating, and bending behaviours of viscous fluids with liquid-air free surfaces, even for when viscosity is constant throughout the liquid. A unified pressure-velocity coupled Stokes solver was developed by Larionov et al. [16] which follows a similar variational principle to correctly enforce the true pressure-velocity coupled zero-traction free surface boundary condition, yielding slightly more realistic coiling effects compared to the Full Form Viscosity, albeit a much larger SPD system involving all velocity components along with pressure must be solved simultaneously.

Accelerated methods for increased efficiency in Eulerian viscosity. The three main time-implicit Eulerian viscosity models have been introduced. From most to least efficient as well as least to most compelling viscous fluid behaviour, we have Laplacian Viscosity, Full

Form Viscosity, and the pressure-velocity coupled unsteady Stokes solver of Larionov et al. [16]. Considering that the latter two methods give compelling viscous phenomena, along with the Full Form Viscosity being the most widely adopted in commercial software and not having an extremely large discrepancy in results compared to the unsteady Stokes solver, we now discuss acceleration methods which aim to increase the efficiency of these two models. Eulerian Viscosity involving spatial adaptivity applied to the Full Form Viscosity Equation 2.6 to yield dimension reduction has been implemented by Batty and Houston [5] via adaptive tetrahedral meshes and Goldade et al. [13] via octrees in a respective finite volume and finite difference setting. Aanjaneya et al. [2] design a multigrid preconditioner combined with a conjugate gradient (CG) solver to increase the viscosity solve speed of the Full Form Viscosity. Panuelos et al. [21] use incompressible polynomial vector fields to represent interior regions of the fluid to accelerate the unsteady Stokes solver.

Accelerated methods for Pressure Projection. Literature involving accelerating the operator-split pressure Equations 2.7, which equivalently yield a Poisson problem, will now be discussed. Adaptive methods here may be applicable to viscosity since both involve solving a SPD linear system of similar stencil nature. Spatial adaptivity via octrees have long been examined in the pressure projection context, beginning in the work of Popinet [22]. Losasso et al. [17] as well as Shi and Yu [24] extend this to SPD formulations for liquid and smoke. Tetrahedral meshes were utilized in conjunction with an algebraic multigrid solver to address the Poisson problem in Chentanez et al. [8]. Goldade et al. [12] introduce a surface-only approach to adjust the pressure projection to achieve realistic immersed bubbles interacting with fluid while also developing multigrid-based preconditioners for CG. They further use spatial adaptivity via affine vector fields, similar to Panuelo et al. [21] who built upon this framework for dimension reduction to accelerate the pressure solve. Aanjaneya et al. [1] use power diagrams and octrees along with a multigrid matrix preconditioner to accelerate the Poisson problem.

2.2 Mathematical Background

2.2.1 Navier-Stokes Fluid Equations

The incompressible Navier-Stokes fluid equations are

$$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \tau + \frac{1}{\rho}\vec{f} \tag{2.1}$$

$$\nabla \cdot \vec{u} = 0 \tag{2.2}$$

$$\tau = \mu \left(\nabla \vec{u} + (\nabla \vec{u})^T\right) \tag{2.3}$$

where $\vec{u}$ is the velocity field, μ is the viscosity coefficient, p is the fluid pressure, ρ is the fluid density, $\vec{f}$ is external body forces (e.g. gravity), and τ is the viscous shear stress tensor. Equation 2.1 is essentially $a = \frac{F_{\text{net}}}{m}$ for a continuum fluid, taking into account advection forces, pressure forces, viscous forces, and external body forces. Equation 2.2 is a continuity equation referred to as the incompressibility condition, ensuring that the mass of the fluid is conserved.

Operator splitting is a very common strategy used to numerically solve these equations in computer graphics, and we adopt this approach in this project and focus on editing the viscosity step. The full set of equations to solve after operator splitting are

$$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u} \tag{2.4}$$

$$\frac{\partial \vec{u}}{\partial t} = \frac{1}{\rho}\vec{f} \tag{2.5}$$

$$\frac{\partial \vec{u}}{\partial t} = \frac{1}{\rho}\nabla \cdot \left(\mu \left(\nabla \vec{u} + (\nabla \vec{u})^T\right)\right) \tag{2.6}$$

$$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho}\nabla p \quad s.t. \quad \nabla \cdot \vec{u} = 0. \tag{2.7}$$

A standard operator-split numerical simulation involves the following: A time step Δt is chosen, adhering to the Courant-Friedrichs-Lewy (CFL) condition for numerical stability. An initial divergence-free velocity field gets advected forward by Δt by integrating Equation 2.4, then gravity plus any other body forces, viscosity, and finally the pressure projection are sequentially integrated forward by Δt via solving Equation 2.5, Equation 2.6, and Equation 2.7 respectively with each respective output velocity being the initial input velocity of the next equation. The final pressure projection in Equation 2.7 leaves the resultant velocity field divergence-free for the subsequent advection step in the next time step.

This project builds upon Eulerian fluid simulations in which quantities are tracked at fixed locations in space [6], as opposed to tracking specific particle quantities in the Lagrangian viewpoint. We further adopt the widely-used marker-and-cell (MAC) grid introduced by Harlow and Welch [14] which employs a staggered grid for storing velocity components, pressure, and other fluid quantities at different locations. In two dimensions

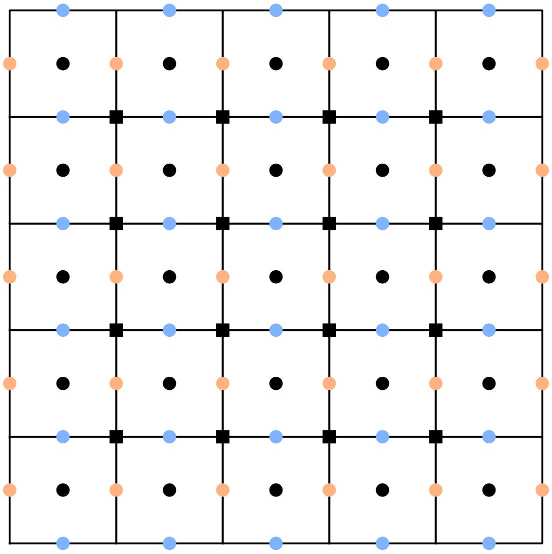


Figure 2.1: A 2D staggered MAC grid for uniform grid resolution $N = 5$ with u velocities stored at shaded orange circles, v velocities at shaded blue circles, pressure values at shaded black circles, and off-diagonal discretized τ locations at shaded black squares.

a staggered MAC grid has pressure values at the centre of the cell, while the horizontal u-component velocity is sampled at the centers of the vertical cell faces and the vertical v-component velocity is sampled at the centers of the horizontal cell faces. This gives two u-velocity values to the left and right of each pressure value, and two v-velocity values below and above of each pressure value, making central differences robust in the spatial discretizations presented later on. See Figure 2.1 for a MAC grid example. This project focuses on alternative ways to increase the efficiency of viscosity integration, specifically Equation 2.6 and so only a brief mention of the most standard methods surrounding the other non-viscosity components of the full fluid solver will be presented here. Equation 2.4 is the advection equation which is usually solved with a semi-Lagrangian procedure [6]. Body forces are traditionally integrated via explicit methods [6] then added to the intermediate velocity value obtained after solving the advection equation before continuing onward with the viscosity solve. An in-depth overview on numerically integrating the viscosity Equation 2.6 is outlined in Section 2.2.3. Pressure projection in Equations 2.7

is commonly solved by plugging the first equation of Equation 2.7 into the second to obtain the associated Poisson problem $\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot \vec{u}$. This Poisson equation is implicitly integrated with finite difference or finite volume methods [6, 3, 23, 19].

2.2.2 Viscosity Boundary Conditions

The remainder of this chapter focuses on viscosity integration of Equation 2.6 after operator-splitting the Navier-Stokes fluid equations. The true zero-traction free surface condition at the liquid-air interface is

$$(-pI + \tau)\vec{n} = 0 \tag{2.8}$$

with the identity matrix I and the surface normal vector $\vec{n}$. In the absence of surface tension, this condition enforces no traction on the plane of the surface. This pressure-velocity coupled boundary condition was only originally solved recently via a pressure-velocity coupled stokes solver developed by Larionov et al. [16]. Enforcing Condition 2.8 is what gives the known viscous fluid behaviours such as viscous buckling, rotation, and coiling.

The operator splitting approach we adopt in this document solves viscosity separately from the pressure projection. The free surface boundary condition is thus also split between the separate solvers, yielding a simplified zero-traction free surface condition that only the shear stress tensor applied to the free surface normal vector need be zero during the viscosity solve

$$\tau\vec{n} = 0 \tag{2.9}$$

while the pressure free surface condition in the separate pressure projection is that the pressure must be zero at the free surface ($pI = 0$). This decoupled operator-split simplification of the free surface condition still yields convincing and compelling viscous behaviour such as buckling and rotational movement, while maintaining desirable efficiency benefits from solving smaller systems of equations compared to the pressure-viscosity coupled system in the Stokes solver of Larionov et al. [16]. Further, this decoupled approach is widely adopted and used in commercial fluid animation software such as SideFX Houdini [25].

Solid boundary conditions are not the focus of this project, yet are quite simple with $\vec{u} = \vec{u}_{\text{solid}}$ being the no-slip Dirichlet condition [6].

2.2.3 Viscosity Discretization of Time and Space

To solve Equation 2.6 numerically, choices must be made regarding the spatial and temporal discretization schemes. A standard first order time-implicit method such as Backwards

Euler gives numerical stability irrespective of the time-step chosen, Δt, and is much more stable compared to explicit methods such as Forward Euler which limits this time step size. Higher order implicit methods can be used, but are not as common as implicit Euler for simulating fluids in computer graphics and so we choose to use this temporal discretization scheme.

The semi-discretized version of Equation 2.6 according to Implicit Euler is

$$\vec{u} = \vec{u}^{old} + \frac{\Delta t}{\rho} \nabla \cdot \left(\mu \left(\nabla \vec{u} + (\nabla \vec{u})^T\right)\right) \tag{2.10}$$

where $\vec{u}^{old}$ is the velocity field directly before the viscosity solve, while $\vec{u}$ denotes the new, final velocity field which results after the viscosity solve. We now seek a spatial discretization which can solve Equation 2.10 while also enforcing the free surface boundary condition 2.9. One can choose to solve the first while then separately dealing with the complex boundary condition, or one can follow the insights and simple yet robust scheme of Batty and Bridson [4] which employs a variational interpretation of viscosity to solve Equation 2.10, while also automatically enforcing the complex free surface boundary condition by construction. We use this variational interpretation and derive the corresponding stencil and matrix system now. Following Batty and Bridson [4], the variational interpretation of viscosity seeks an unknown velocity field which minimizes the following energy functional $E_1[\vec{u}]$ with respect to the unknown velocity field

$$E_1[\vec{u}] = \iiint_\Omega \left(\frac{\rho}{2\Delta t} \left\| \vec{u} - \vec{u}^{old} \right\|^2 + \mu \left\| \frac{\nabla \vec{u} + (\nabla \vec{u})^T}{2} \right\|_F^2 \right) dV \tag{2.11}$$

where $|| \ ||_F$ denotes the Frobenius norm, Ω is the fluid domain, and dV is a differential fluid volume. $E_1[\vec{u}]$ is both convex and quadratic in $\vec{u}$, and after discretizing and minimizing we end up with symmetric positive definite (SPD) system which is conducive to solving efficiently with iterative linear matrix solvers such as a preconditioned CG algorithm. Discretizing $E_1[\vec{u}]$ in space with finite differences and sums is the next step. In two dimensions instead of three for simplicity, we have $\vec{u} = \begin{bmatrix} u \\ v \end{bmatrix}$ storing the x-component and y-component velocities u and v, and the above volume integral becomes an area integral over the 2D fluid domain with differential area dA. For a 2D square MAC grid arranged as in Figure 2.1 with uniform grid resolution $N = N_x = N_y$ giving $\Delta x = \Delta y$, the integral can be approximated with the following discrete sums and central finite differences for partial derivatives:

$$\begin{aligned}\iint_{\Omega} \rho \left\| \vec{u} - \vec{u}^{old} \right\|^2 dA &= \iint_{\Omega} \rho \left((u - u^{old})^2 + (v - v^{old})^2 \right) dA \\ &\approx \sum_{i,j \in \Omega} V^u_{i+\frac{1}{2},j} \rho_{i+\frac{1}{2},j} \left(u_{i+\frac{1}{2},j} - u^{old}_{i+\frac{1}{2},j} \right)^2 \Delta x^2 \\ &\quad + \sum_{i,j \in \Omega} V^v_{i,j+\frac{1}{2}} \rho_{i,j+\frac{1}{2}} \left(v_{i,j+\frac{1}{2}} - v^{old}_{i,j+\frac{1}{2}} \right)^2 \Delta x^2\end{aligned}$$

$$\begin{aligned}\iint_{\Omega} \mu \left\| \frac{\nabla \vec{u} + (\nabla \vec{u})^T}{2} \right\|_F^2 dA &= \iint_{\Omega} \mu \left\| \begin{bmatrix} u_x & \frac{u_y + v_x}{2} \\ \frac{u_y + v_x}{2} & v_y \end{bmatrix} \right\|_F^2 dA = \iint_{\Omega} \mu \left(u_x^2 + v_y^2 + \frac{(u_y + v_x)^2}{2} \right)_F^2 dA \\ &\approx \sum_{i,j \in \Omega} V^p_{ij} \mu_{ij} \left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} \right)^2 \Delta x^2 + \sum_{i,j \in \Omega} V^p_{ij} \mu_{ij} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x} \right)^2 \Delta x^2 \\ &\quad + \sum_{i,j \in \Omega} V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}} \mu_{i+\frac{1}{2},j+\frac{1}{2}} \frac{\left(\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta x} + \frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x} \right)^2}{2} \Delta x^2\end{aligned} \tag{2.12}$$

The above involves a spatially varying viscosity μ_{ij} which is piecewise constant as well as the crucial volume fraction arrays V^u, V^v, V^p, $V^{\tau^{12}}$ being the key to enforcing the traction-free free surface boundary condition 2.9. These volume fractions (area fractions in 2D) vary across the domain and denote the fraction of liquid inside a control volume, with one being completely filled with liquid and zero being completely empty. Location-wise in 2D, $V^u_{i+\frac{1}{2},j}$, $V^v_{i,j+\frac{1}{2}}$, V^p_{ij}, and $V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}$ are $\Delta x \times \Delta x$ area squares with inner midpoints respectively populated at $u_{i+\frac{1}{2},j}$, $v_{i,j+\frac{1}{2}}$, p_{ij}, and $\tau^{12}_{i+\frac{1}{2},j+\frac{1}{2}}$ quantity locations as in Figure 2.2. Note that V^p equivalently holds the volume fractions for the diagonal stress components τ_{11} and τ_{22} since these are located at the same location as pressure values. We choose to follow similar notation to Batty and Bridson [4] regarding these volume fractions.

The above discretized sums support spatially variable viscosity. In the following contexts we confine ourselves to a fixed viscosity coefficient $\tilde{\mu}$ throughout the domain since we later combine this method with the Laplacian Viscosity formulation which can not support variable viscosity. Discretizing then minimizing the integral $E_1[\vec{u}]$ via differentiating with respect to $u_{i+\frac{1}{2},j}$ and then equating to zero gives the following implicit update for the x-component velocity in 2D with fixed viscosity constant $\tilde{\mu}$ for the Full Form Viscosity:

$$\begin{aligned} u_{i+\frac{1}{2},j} + \frac{\Delta t\ \tilde{\mu}}{\Delta x^2(\rho V^u)_{i+\frac{1}{2},j}} \Bigg(&\left(2V^p_{ij} + 2V^p_{i+1,j} + V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}} + V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\right) u_{i+\frac{1}{2},j} - 2V^p_{ij}u_{i-\frac{1}{2},j} \\ &- 2V^p_{i+1,j}u_{i+\frac{3}{2},j} - V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}u_{i+\frac{1}{2},j+1} - V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}u_{i+\frac{1}{2},j-1} \\ &+V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}(v_{i,j+\frac{1}{2}} - v_{i+1,j+\frac{1}{2}}) + V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}(v_{i,j-\frac{1}{2}} - v_{i+1,j-\frac{1}{2}})\Bigg) \\ &= u^{old}_{i+\frac{1}{2},j} \end{aligned} \tag{2.13}$$

See Appendix A with $\alpha = 1$ for a detailed derivation of Equation 2.13. The equation for $v_{i,j+\frac{1}{2}}$ is of a very similar form. Equation 2.13 is a nine-point stencil in 2D and we see that the u and v velocity components are coupled to one another, yielding one SPD system to solve for both u and v degrees of freedom simultaneously. Multiplying Equation 2.13 through by $(\rho V^u)_{i+\frac{1}{2},j}$ the resultant SPD system can be expressed as

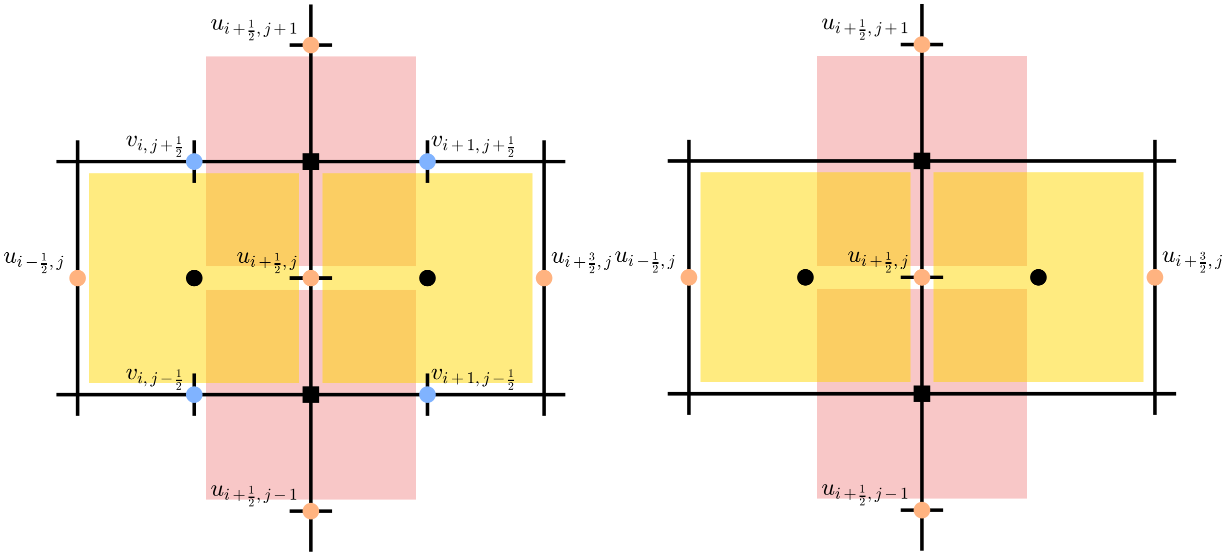
$$(W_{V^{uv}}W_\rho + 2\Delta t\ \tilde{\mu}D^TKW_{V^\tau}D)\vec{u} = W_{V^{uv}}W_\rho\vec{u}^{old} \tag{2.14}$$

where $\vec{u}$ here stores all discrete u and v degrees of freedom in a lexicographic order, each W is a diagonal matrix holding the corresponding volume fractions ($W_{V^{uv}}$, W_{V^τ}) or fluid density (W_ρ), D is a discretized linear operator such that $D\vec{u} \approx \frac{\nabla\vec{u}+(\nabla\vec{u})^T}{2}$ and K is a diagonal matrix which doubles the contribution of the off-diagonal stress components τ^{12} since off-diagonal components contribute twice to the Frobenius norm from the symmetric form of the shear stress tensor ($\tau^{12} = \tau^{21}$). Note that W_{V^τ} contains both the $V^{\tau^{12}}$ and V^p volume fraction, while $W_{V^{uv}}$ holds both V^u and V^v volume fractions.

Under the assumption of a constant viscosity throughout the liquid, $\tilde{\mu}$, the divergence operator in Equation 2.6 can be distributed to both terms, giving a $\nabla\cdot(\nabla\vec{u})^T$ term which can then be re-expressed as $\nabla(\nabla\cdot\vec{u})$ which simply vanishes under the incompressibility constraint $\nabla\cdot\vec{u} = 0$. This simplifies the Full Form Viscosity Equation 2.6 to the Laplacian Form of viscosity

$$\frac{\partial\vec{u}}{\partial t} = \frac{\tilde{\mu}}{\rho}\nabla^2\vec{u} \tag{2.15}$$

which is equivalent to a re-scaled heat equation for each velocity component. This Laplacian Form of viscosity was introduced to computer graphics by Stam [15] and the first implicit formulation with free surfaces was implemented by Carlson et al. [7]. The different components of velocity are decoupled from one another from having no cross terms



Full Form Stencil

Laplacian Form Stencil

Figure 2.2: 2D Stencil partners for a general discrete x-velocity DOF $u_{i+\frac{1}{2},j}$. For the discretized shear stress tensor, diagonal components such as τ_{ij} and $\tau_{i+1,j}$ are located at the left and right black shaded circles, while off-diagonal nodal stress components such as $\tau_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\tau_{i+\frac{1}{2},j-\frac{1}{2}}$ are located at cell corners at the upper and lower shaded black squares. Note that p_{ij} and $p_{i+1,j}$ are also located at τ_{ij} and $\tau_{i+1,j}$. Corresponding control volumes for the volume fractions V^p and $V^{\tau^{12}}$ are shaded yellow and red, centred at their respective quantity. Control volumes for volume fractions V^u and V^v are centred at u and v DOFs.

from the vanished $(\nabla\vec{u})^T$ term, and thus it is much less costly to solve compared to the velocity-coupled Full Form Viscosity. The Laplacian Form has a five-point neighbor stencil for each separate system in both 2D and 3D, while the Full Form Viscosity has a nine-point stencil for the 2D coupled system and a 13-point stencil for the 3D coupled system. The semi-discrete form of Equation 2.15 according to implicit Euler is

$$\vec{u} = \vec{u}^{old} + \frac{\Delta t\ \tilde{\mu}}{\rho}\ \nabla^2\vec{u} \tag{2.16}$$

and the equivalent energy functional to this semi-discrete form is

$$E_0[\vec{u}] = \iiint_\Omega \left(\frac{\rho}{\Delta t}\left\|\vec{u}-\vec{u}^{old}\right\|^2 + \mu\left\|\nabla\vec{u}\right\|_F^2\right) dV \tag{2.17}$$

which gives the incorrect natural free surface boundary condition from minimizing, $\nabla\vec{u}\, n = 0$. Again, following the derivation in Appendix A with $\alpha = 0$, after discretizing then minimizing we get the implicit Laplacian Viscosity update for the x-component velocity

$$\begin{aligned} u_{i+\frac{1}{2},j} + \frac{\Delta t\ \tilde{\mu}}{\Delta x^2(\rho V^u)_{i+\frac{1}{2},j}} \Bigg(& \left(V^p_{ij} + V^p_{i+1,j} + V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}} + V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\right) u_{i+\frac{1}{2},j} - V^p_{ij} u_{i-\frac{1}{2},j} \\ & -V^p_{i+1,j} u_{i+\frac{3}{2},j} - V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+1} - V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-1} \Bigg) \\ & = u^{old}_{i+\frac{1}{2},j} \end{aligned} \tag{2.18}$$

which has no coupling to the y-velocity v and yields an independent SPD linear system for the u degrees of freedom (DOFs). See Figure 2.2 for the u stencil. A similar stencil and SPD system arises for the v DOFs. The matrices which arise are M-matrices with the real part of all eigenvalues being positive. This property can increase the efficiency of preconditioned CG. The exact form of the matrices are below

$$(W_{V^u} W_\rho + \Delta t\ \tilde{\mu} W_{V^\tau} M)u = W_{V^u} W_\rho u^{old}, \qquad (W_{V^v} W_\rho + \Delta t\ \tilde{\mu} W_{V^\tau} M)v = W_{V^v} W_\rho v^{old}$$

with M being the discrete Laplacian operator matrix. Theses systems are identically diffusion systems up to a scaling.

This research project seeks to develop a mixed discretization scheme to (1) exploit efficiency in the interior of the liquid by implementing an efficiently solved Partial Differential Equation (PDE) 2.15 throughout a potentially large portion of the interior and (2) employ the more accurate PDE 2.6 near the liquid-air surface of the fluid. Adopting the latter allows our method to preserve and satisfy the zero-traction free surface boundary condition 2.9 which is necessary to giving convincing viscous fluid behaviours from viscous forces near the liquid surface.

Chapter 3

Methodology

In this chapter, we delineate avenues explored for a mixed viscosity discretization to integrate the less costly Laplacian Viscosity Equation 2.15 in the vast interior of the fluid, while integrating the more costly Equation 2.6 near the air free surface to get the correct boundary conditions. We propose three novel mixed discretization coupling approaches to do so. The first approach uses the corresponding stencil for whichever region the DOF is located in, the second approach treats the inner discretization split boundary as a pseudo free surface, and the third combines and minimizes a mixed energy combination between the Laplacian and Full Form energies $E_0[\vec{u}]$ and $E_1[\vec{u}]$ in different regions of space. We describe these approaches in Sections 3.1.1 – 3.1.3. We further evaluate each novel discretization's ability to solve the two PDEs in their respective regions in the subsequent Section 4.1 via a mixed spatial convergence test of the viscosity step alone. We then move forward with plugging two of these mixed viscosity discretizations into a full 2D fluid solver in the subsequent Section 4.2 and evaluate computational costs as well as the corresponding qualitative viscous fluid behaviours.

3.1 Mixed Discretizations Explored

Consider a region of liquid whose boundary consists of a free surface and (optionally) a solid boundary, as depicted in Figure 3.1. We divide the space into the outer air region, Ω_{air}, an outer layer of liquid Ω_F, and an inner region of liquid Ω_{∇^2}. We refer to the inner liquid boundary between Ω_F and Ω_{∇^2} as Γ_D.

Given these domains, we wish to apply the Laplacian Viscosity discretization in Ω_{∇^2}, the Full Form Viscosity in Ω_F, and determine a strategy for coupling the two together across Γ_D.

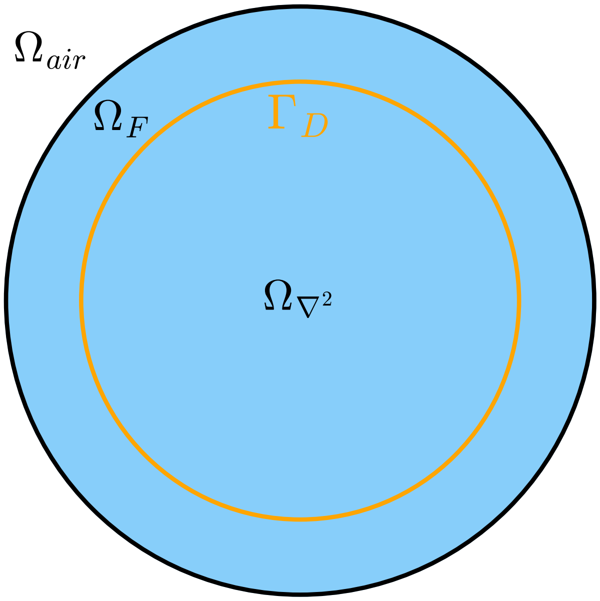


Figure 3.1: Mixed domain example set-up for a circular liquid. The Laplacian Viscosity is implemented in the interior Ω_{∇^2} region while the Full Form Viscosity is implemented in Ω_F to handle the liquid-air free surface as well as liquid-solid boundary interactions.

3.1.1 Per-DOF Stencil-Selection

The first strategy we propose for constructing a mixed discretization is to respectively use stencil Equation 2.13 for the DOFs in Ω_F and stencil Equation 2.18 for DOFs in Ω_{∇^2}. A consequence from using this treatment is that the resultant sparse matrix is no longer SPD as we have lost symmetry from asymmetric stencil arms interacting with one another across the Γ_D boundary. Preconditioned CG solvers can no longer be used, and we must resort to preconditioned Krylov subspace solvers like the Generalized Minimal Residual Method (GMRES) or the Bi-Conjugate Gradient Stabilized method (BiCGStab).

3.1.2 Free Surface Mixed Approach

The second method we propose is one which treats the inner liquid discretization boundary Γ_D as a free surface, with air on the other side, from each discretization domain's respective point of view. It should be noted that the separate methods are not using their own DOFs — Each discretization makes use of the *same* DOFs in the region near the surface, so those DOFs receive forces from both discretizations, and we recover a single continuous velocity field.

A matrix $A_{\Omega_{\nabla^2}}$ is constructed from minimizing Equation 2.17 in a liquid-filled Ω_{∇^2} region with a free surface located at Γ_D, assuming that both Ω_F and Ω_{air} contain air.

A matrix A_{Ω_F} is constructed from minimizing Equation 2.11 in a liquid-filled Ω_F region with a free surface located at Γ_D, assuming that both Ω_{∇^2} and Ω_{air} contain air.

Since minimizing Equations 2.17 or 2.11 on their own with free surfaces yield SPD systems, both $A_{\Omega_{\nabla^2}}$ and A_{Ω_F} are SPD matrices. We now add both together to get

$$A = A_{\Omega_{\nabla^2}} + A_{\Omega_F} \tag{3.1}$$

which is also SPD. Matrix A is the resultant Free Surface Mixed discretization matrix used in the viscosity solve for this method. Since A is symmetric, preconditioned CG can be used to solve it.

3.1.3 Smoothing-α Mixed Approach Between $E_1[\vec{u}]$ and $E_0[\vec{u}]$

This third mixed method seeks to minimize a combination of the two energies corresponding to the Full Form Viscosity and Laplacian Viscosity, respectively Equation 2.11 and Equation 2.17, at different locations in space. Specifically, the combined energy to minimize is

$$E_\alpha[\vec{u}] = \iiint_\Omega \left(\frac{\rho}{2\Delta t} \left\| \vec{u} - \vec{u}^{old} \right\|^2 + \alpha \left(\mu \left\| \frac{\nabla\vec{u} + (\nabla\vec{u})^T}{2} \right\|_F^2 \right) + \frac{(1-\alpha)}{2} \left(\mu \left\| \nabla\vec{u} \right\|_F^2 \right) \right) dV \tag{3.2}$$

with $\alpha \in [0..1]$ being piecewise constant, varying in space, and corresponding to different control volumes in space just like volume fractions. See Appendix A for the minimization of Equation 3.2 where we similarly discretize and define α analogously to volume fractions. $E_\alpha[\vec{u}]$ is clearly equivalent to $E_1[\vec{u}]$ in Equation 2.11 for when all α array values are 1, and equivalent to $E_0[\vec{u}]$ in Equation 2.17 for when all α array values are 0.

An α value of zero corresponds to a pure Laplacian Viscosity control volume, while an α value of one corresponds to a pure Full Form Viscosity control volume, with $\alpha \in (0, 1)$ a combination of the two. Since $E_0[\vec{u}]$ and $E_1[\vec{u}]$ are convex and quadratic in $\vec{u}$, adding a combination of the two also yields a convex and quadratic function and thus $E_\alpha[\vec{u}]$ yields a desirable property — a SPD linear system to solve with preconditioned CG. This method is actually a more general case of the previous Free Surface Mixed Method introduced in Section 3.1.2 which involved a sharp transition of the α arrays from one to zero at Γ_D while moving deeper into the liquid. We implemented both gradual and sharp transitions of α in space via letting α be determined by a smoothstep function.

For example, in the mixed spatial convergence test results in Table 4.1 with a vertical line splitting the domain into two distinct discretization areas Ω_{∇^2} and Ω_F as in Scenario (1) of Figure 4.2, $\alpha = \alpha(x)$ is a simple scalar function equivalent to a translated smoothstep(x) function which we chose to transition from zero to one from $x = [\frac{\pi}{4}, \frac{3\pi}{4}]$:

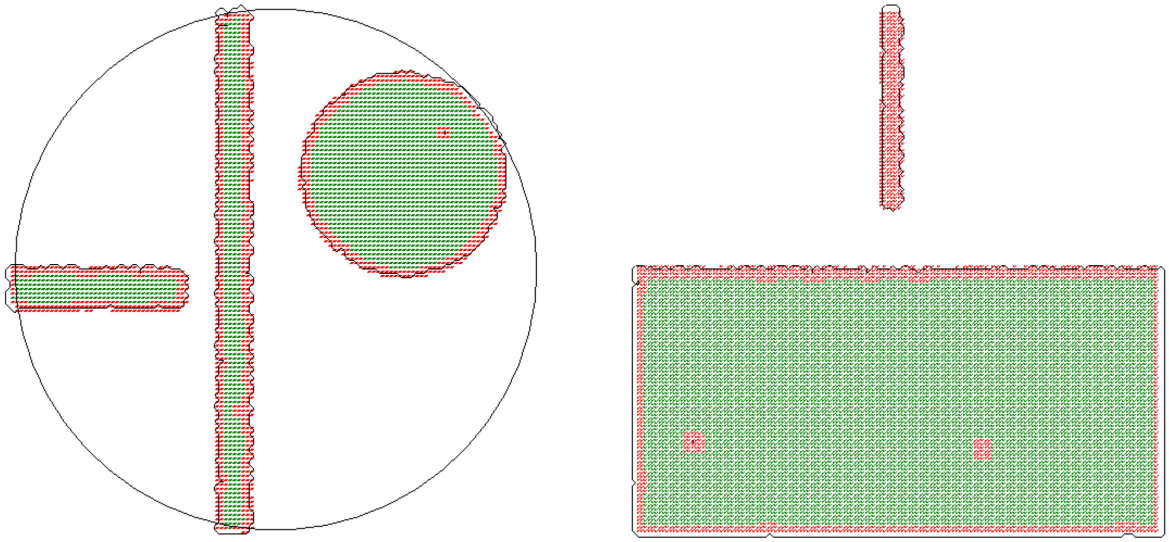
$$\alpha(x) = \begin{cases} 0 & \text{if } x \leq \frac{\pi}{4}, \\ 3\left(\frac{x-\frac{\pi}{4}}{\frac{3\pi}{4}-\frac{\pi}{4}}\right)^2 - 2\left(\frac{x-\frac{\pi}{4}}{\frac{3\pi}{4}-\frac{\pi}{4}}\right)^3 & \text{if } \frac{\pi}{4} < x < \frac{3\pi}{4}, \\ 1 & \text{if } x \geq \frac{3\pi}{4}. \end{cases} \tag{3.3}$$

In the mixed spatial convergence test we also tested cases in which the smoothstep(x) transitions from zero to one from $x = [0, \pi]$ as well as $x = [\frac{9\pi}{20}, \frac{11\pi}{20}]$.

3.2 Inner Liquid Discretization Boundary Γ_D

This section explains a simple low-overhead algorithm to define the discretization splitting boundary Γ_D to divide the inner-liquid discretization regions Ω_F and Ω_{∇^2}. We refer to this algorithm as the Square-radius Neighbor Liquid Content algorithm. This algorithm labels a DOF as an interior Ω_{∇^2} DOF if all of the neighboring cells (within a local neighborhood) are fully filled with fluid. In two dimensions (for simplicity) we define the local neighborhood to be a square region surrounding the DOF. We thus define a distance parameter d which is the half square 'radius' length to look in all directions, and then check if all the DOFs contained within the surrounding square of side length $2d$ are fully filled with liquid via examining the volume fractions.

For the Per-DOF Stencil Method of Section 3.1.1, we do the following. If all the volume fractions within the surrounding square are one (fully liquid), then this DOF is inside Ω_{∇^2}. Otherwise, the DOF is *closer* to the air-liquid surface and we label it a Ω_F DOF. The u



Scene 1 of Section 4.2

Scene 2 of Section 4.2

Figure 3.2: Visual depiction of the Square-radius Neighbor Liquid Content Algorithm in Section 3.2 which is used to label DOFs as either Laplacian Ω_{∇^2} DOFs (green) or Full Form Ω_F DOFs (red). These are the initial condition frames for the scenes of Section 4.2. Scene 1 uses $N = 180$ and Scene 2 uses $N = 150$.

DOFs only check nearby u DOF volume fractions V^u, while the v DOFs only check nearby v DOF volume fractions V^v.

For the Free Surface Mixed Method of Section 3.1.2, we initially construct A_{Ω_F} by letting the u DOF control volume fractions V^u within Ω_{Ω_F} be one, and then we construct a pseudo free boundary at Γ_D by estimating the the other volume fractions V^v, V^p, and $V^{\tau^{12}}$ from this V^u. The volume fractions are then inverted to get the appropriate volume fractions to then construct $A_{\Omega_{\nabla^2}}$.

Chapter 4

Implementation and Results

We propose a mixed spatial convergence test to evaluate each mixed discretization's ability to solve the different PDEs in the respective inner-liquid sub-domains. From there, we move forward with implementing two of the mixed discretization schemes from Sections 3.1.1 and 3.1.2 into full 2D fluid solvers and evaluate (i) computational cost of each against the Full Form Viscosity benchmark and Laplacian Viscosity benchmark for different iterative solvers as well as (ii) qualitative visual comparisons of each mixed method against the ideal benchmark viscous fluid behaviour of the Full Form Viscosity.

4.1 Mixed Spatial Convergence Test Results

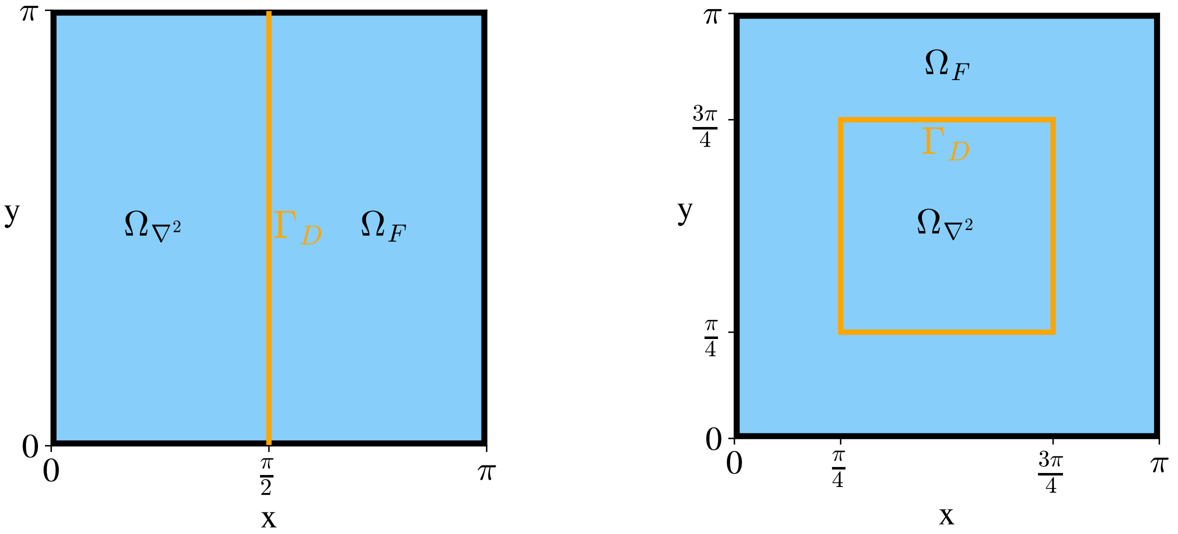
Although visual results and speed are often prioritized in computer graphics over spatial accuracy, we nonetheless focus our attention on numerical accuracy in this section for each proposed mixed method. This is motivated by having a quantitative measure of the effectiveness and accuracy of the mixed methods to solve Equation 2.15 in certain locations of the domain, while also accurately solving Equation 2.6 in the other portions of the domain.

Following the same two 2D test cases in the work of Wang [27] along with a 3rd case, we now formulate a *mixed* discretization test in which we (i) initialize one portion of the domain Ω_F with the Laplacian velocity input, while using the Full Form input in the rest of the domain Ω_{∇^2}. We then employ each mixed discretization method in its respective domain region and then compare to the analytical exact solution. This is done for different grid resolutions of the a uniform MAC grid, and the infinity norm error convergence is examined for each mixed method.

All three 2D test cases are conducted in a $\pi \times \pi$ square domain filled with fluid, with outer Dirichlet solid boundaries of the box which are zero and with $N = N_x = N_y$ for simplicity. We also simply set Δt, ρ μ to be one as in Wang [27]. These tests are solely applied to the viscosity step, not the full fluid solver. Two mixed domain scenarios are examined for each test case:

1. **Line** split: A vertical discretization divider line is placed down the direct centre of the domain, separating region Ω_F on the left half and Ω_{∇^2} on the right half.
2. **Square** split: An internal square region in the direct centre of the domain of size $\frac{\pi}{2} \times \frac{\pi}{2}$ is region Ω_{∇^2}, while the region external to this interior square is region Ω_F.

Scenario (1) is examined for simplicity, while Scenario (2) more appropriately reflects the actual use case of this mixed method since the Laplacian region Ω_{∇^2} only interacts inside the fluid and only has a boundary with the Ω_F region. Scenario (2) further has right angles separating Ω_F and Ω_{∇^2} to test corners in a 2D setting.



Scenario (1): **Line** split

Scenario (2): **Square** split

Figure 4.1: 2D Mixed Convergence test scenarios for the mixed domain input.

Each test case is derived via beginning from an analytical exact velocity field solution at $t = \Delta t$ which has outer Dirichlet solid boundaries which are zero. The initial velocity input is then found via analytically integrating backwards the semi-discretized form for each discretization, Equations 2.10 and 2.16, by Δt to have the respective inputs for the Ω_F and Ω_{∇^2} regions. For input regions in which there exists coincident DOFs associated with *both* discretizations at the boundary between Ω_F and Ω_{∇^2}, we simply take the average of both discretization inputs at those locations. This is relevant for the mixed methods Free Surface and Smoothing-α Mixed Methods. The full mixed velocity field input is then used as the initial velocity field which is stepped forward by the discrete mixed method to get an approximate solution to compare with the analytical exact solution.

Test Case I

From Wang [27], this simplest test case is derived by starting from the exact final velocity field at $t = \Delta t$ which is

$$\begin{aligned} u(x, y, t = \Delta t) &= \sin(x)\sin(y) \\ v(x, y, t = \Delta t) &= \sin(x)\sin(y) \end{aligned}$$

giving initial inputs $\vec{u}_F = \begin{bmatrix} u_F \\ v_F \end{bmatrix}$ for region Ω_F and $\vec{u}_{\nabla^2} = \begin{bmatrix} u_{\nabla^2} \\ v_{\nabla^2} \end{bmatrix}$ for Ω_{∇^2}:

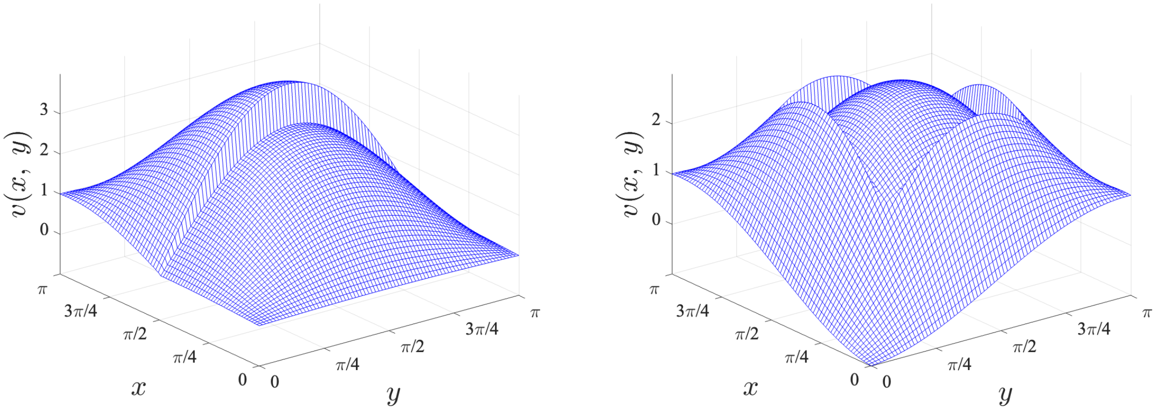
$$\begin{aligned} u_F(x, y, t = 0) &= \left(1 + \frac{3\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y) - \frac{\Delta t\,\mu}{\rho}\cos(x)\cos(y) \\ v_F(x, y, t = 0) &= \left(1 + \frac{3\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y) - \frac{\Delta t\,\mu}{\rho}\cos(x)\cos(y) \\ u_{\nabla^2}(x, y, t = 0) &= \left(1 + \frac{2\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y) \\ v_{\nabla^2}(x, y, t = 0) &= \left(1 + \frac{2\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y) \end{aligned}$$

Test Case II

From Wang [27], this less simple test case is derived via starting from the exact final velocity field at $t = \Delta t$ which is

$$u(x, y, t = \Delta t) = \sin(x)\sin(y) \tag{4.1}$$

$$v(x, y, t = \Delta t) = (x^2 - \pi x)(y^2 - \pi y) \tag{4.2}$$



Scenario (1): **Line** split input

Scenario (2): **Square** split input

Figure 4.2: Mixed input y-component velocity component v for Test Case I for both scenarios (1) and (2) for domain splitting.

giving the initial inputs

$$
\begin{aligned}
u_F(x,y,t=0) &= \sin(x)\sin(y) - \frac{\Delta t\,\mu}{\rho}((2x-\pi)(2y-\pi) - 3\sin(x)\sin(y)) \\
v_F(x,y,t=0) &= (x^2-\pi x)(y^2-\pi y) - \frac{\Delta t\,\mu}{\rho}(\cos(x)\cos(y) + 4(x^2-\pi x) + 2(y^2-\pi y)) \\
u_{\nabla^2}(x,y,t=0) &= \left(1 + \frac{2\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y) \\
v_{\nabla^2}(x,y,t=0) &= (x^2-\pi x)(y^2-\pi y) - \frac{2\Delta t\,\mu}{\rho}((x^2-\pi x) + (y^2-\pi y)).
\end{aligned}
$$

Test Case III

This test case was designed to prove the intuition that Free Surface Mixed method enforces a pseudo zero-traction free surface boundary condition in the *interior* of the liquid at the artificial boundary between Ω_F and Ω_{∇^2}. Let $\psi(x,y) = \sin(x)\sin(y)\cos^2(x)$ for more compact notation in the below. We intuitively choose a final velocity field in which the zero-traction boundary condition is satisfied at $x = \frac{\pi}{2}$:

$$u(x, y, t = \Delta t) = \psi(x, y) \tag{4.3}$$

$$v(x, y, t = \Delta t) = \frac{\psi(x, y)}{\cos^2(x)} \tag{4.4}$$

Under Scenario (1) where we split the domains along a vertical line down the centre of the domain (ie. at $x = \frac{\pi}{2}$), this velocity field clearly satisfies the zero-traction free surface boundary condition 2.9 along this line. We show this now. The normal vector to the interior fluid boundary is $\vec{n} = \hat{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and with $\mu = 1$ and we see that

$$\begin{aligned}
\tau\vec{n} &= \left(\nabla\vec{u} + (\nabla\vec{u})^T\right)\vec{n} \\
&= \begin{bmatrix} 2\partial_x u & \partial_x v + \partial_y u \\ \partial_x v + \partial_y u & 2\partial_y v \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 2\partial_x u \\ \partial_x v + \partial_y u \end{bmatrix} \\
&= \begin{bmatrix} 2\cos(x)\sin(y)\left[\cos^2(x) - 2\sin^2(x)\right] \\ \cos(x)\left[\sin(x) + \cos(x)\sin(x)\cos(y)\right] \end{bmatrix}
\end{aligned}$$

which clearly vanishes for $x = \frac{\pi}{2}$ from the common $\cos(x)$ in both vector components. Once again, analytically integrating the semi-discretized equations we get the initial inputs for this most complicated test case.

$$\begin{aligned}
u_F(x, y, t = 0) &= \left(1 + \frac{\Delta t\,\mu}{\rho}\right)\psi(x, y) - \frac{2\Delta t\,\mu}{\rho}\sin(y)\left(2\sin(x) - \frac{9\psi(x, y)}{\sin(y)}\right) - \frac{\Delta t\,\mu}{\rho}\cos(x)\cos(y) \\
v_F(x, y, t = 0) &= \left(1 + \frac{3\Delta t\,\mu}{\rho}\right)\frac{\psi(x, y)}{\cos^2(x)} - \frac{\Delta t\,\mu}{\rho}\cos(y)\left(3\cos^3(x) - 2\cos(x)\right) \\
u_{\nabla^2}(x, y, t = 0) &= \left(1 + \frac{\Delta t\,\mu}{\rho}\right)\psi(x, y) - \frac{\Delta t\,\mu}{\rho}\sin(y)\left(2\sin(x) - \frac{9\psi(x, y)}{\sin(y)}\right) \\
v_{\nabla^2}(x, y, t = 0) &= \left(1 + \frac{2\Delta t\,\mu}{\rho}\right)\sin(x)\sin(y)
\end{aligned}$$

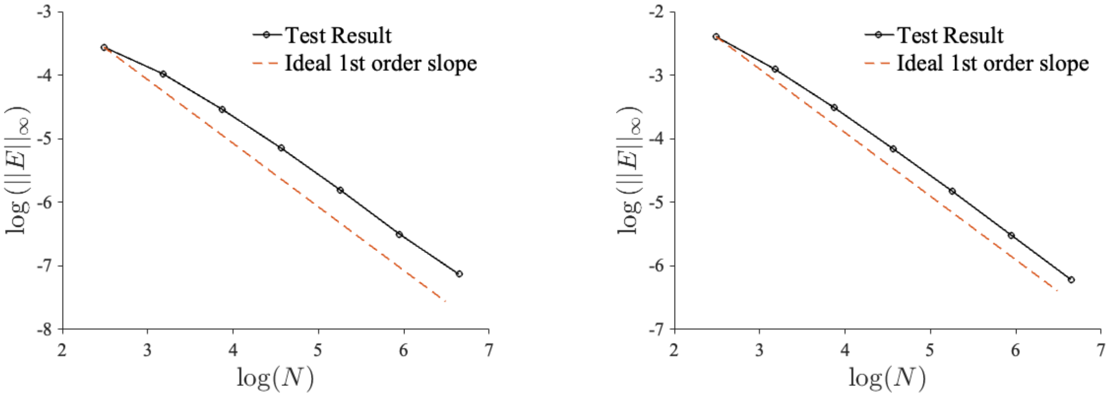
Test Case Results

In all convergence tests presented, we did not employ a second order ghost boundary treatment for the Dirichlet boundaries, so the highest order of convergence possible in these tests is first order convergence of the infinity norm errors for u and v. We thus consider first order convergence of the infinity norm errors as a ‘pass’ (✓) for this convergence test, otherwise a ‘fail’ (✗). A summary of the mixed methods results are presented in Table 4.1. The result that the Free Surface Mixed Method only converges for the Test

	Case 1		**Case 2**		**Case 3**	
	Line	**Square**	**Line**	**Square**	**Line**	**Square**
Per-DOF Stencil 3.1.1	✓	✓	✓	✓	✓	✓
Free Surface 3.1.2	✗	✗	✗	✗	✓	✗
Smoothing-α 3.1.3	✗	✗	✗	✗	✓*	✗

Table 4.1: Mixed Discretization convergence test Pass (✓) and Fail (✗) summary for Scenario (1) and Scenario (2) domains as shown in Figure 4.2. A (✓) indicates first order infinity norm error convergence between exact solution and the mixed discrete solution. A (✗) indicates no convergence.

Case 3 in Table 4.1 shows that the artificial internal free surface implemented from both discretizations indeed (undesirably) enforces the zero-traction boundary condition there, which is the reason the Free Surface Mixed Method fails to converge in all other cases. An example of successful spatial convergence for Test Case 3 is presented in Figure 4.3. The



u velocity

v velocity

Figure 4.3: Log-log plots of the infinity norm error $||E||_\infty$ vs N showing successful first order convergence for Test Case III with Line Scenario (1) for the Free Surface Mixed Method of Section 3.1.2.

Smoothing-α Mixed Method is a more general case of the Free Surface Mixed Method. It fails all tests unless the full α transition zone approaches a single vertical line, in which this mixed method converges towards the Free Surface Method and then similarly converges for Test Case 3, Scenario (1).

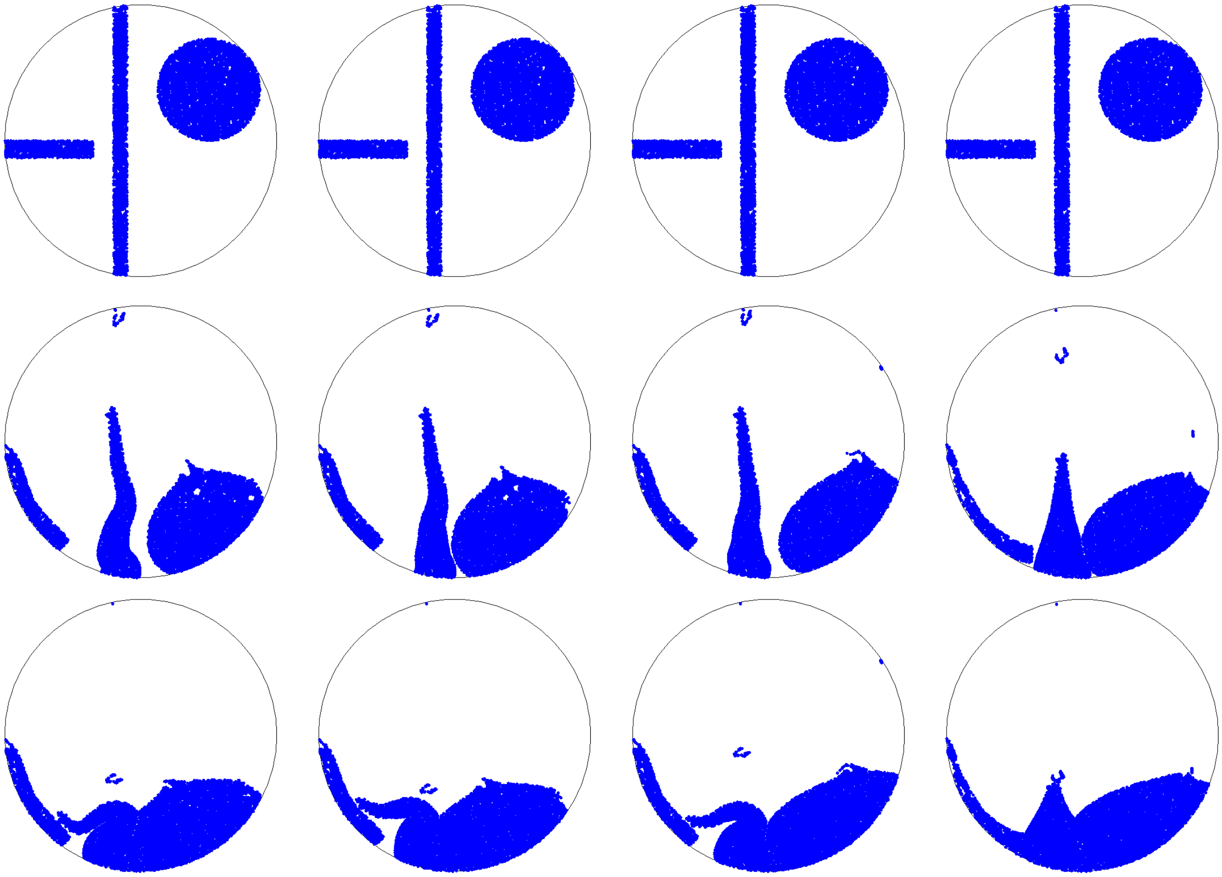
4.2 Full Fluid Solver Results

We implement the proposed mixed methods into full fluid solvers following the operator-split numerical procedure outlined in Section 2.2.1. Other acceleration strategies for Eulerian viscosity have achieved at least first order spatial convergence [13], so we make sure to evaluate the Per-DOF Stencil Mixed method of Section 3.1.1 in this section which was the only mixed method that converged to first order for all six test cases. Although this method increases sparsity of the overall system compared to the Full Form Viscosity benchmark, unfortunately it does not yield a symmetric system. We thus choose to also investigate full fluid solver results with the Free Surface Mixed Method of Section 3.1.2 here, given that it has the desirable property of yielding an SPD system while also increasing the system sparsity. The effectiveness of these mixed discretizations will now be evaluated via computational cost along with qualitative visual comparisons to the Full Form Viscosity and Laplacian Viscosity benchmarks.

A MacBook Air M1 is used for all simulation results. Since we are using operator splitting, we adjust and only change the viscosity step with the mixed methods as a drop-in replacement. The fluid solver used is the C++ open source code from Batty and Bridson [4]. This fluid simulator solves the operator split Navier-Stokes equations with semi-Lagrangian advection as outlined in [6], an external downwards force of gravity via adjusting the y-velocity, the Full Form Viscosity step (which we alter), and then a pressure projection according to Batty et al. [3]. Liquid particle locations are used to track the liquid-air surface, which is then internally converted to a standard level set method for accurate and efficient computation of area liquid fractions [20]. The solid no-slip condition $\vec{u} = \vec{u}_{solid}$ is enforced at solid boundaries and pressure values are set to zero at the free surface to help enforce the natural zero traction operator-split free surface boundary condition 2.9. For the viscosity solve we use various iterative solvers and preconditioners from the ViennaCL and Eigen C++ open source linear algebra libraries, as well as the 'built-in' modified Incomplete Cholesky (ICHOL) CG solver which comes with the open source code base [4]. This built-in modified ICHOL CG solver is used for the pressure projection. Note that as long as a non-zero amount of Full Form DOFs are incorporated within the mixed methods, they lose the property of being an M-matrix and preconditioners for iterative solvers may be less effective.

4.2.1 Scene 1: Circular Boundary with Beam, Column, and Disk

The open source code of Batty and Bridson [4] comes with a scene with a column, beam, and a disk of fluid at rest inside an outer circular solid boundary as in the top row of Figure



Full Form Benchmark Per-DOF Stencil Free Surface Laplacian Benchmark

Figure 4.4: Scene 1 viscous fluid simulation comparison for $N = 150$ at frames [0, 180, 310]. The average NNZ(A) = 91527, 83865, 84727, 68292 respectively for the Full Form Benchmark, Per-DOF Stencil Mixed, Free Surface Mixed, and Laplacian Benchmark.

4.4. These three blobs of fluid fall downwards via a gravitational force and interact with the circular solid boundary. The three qualitative benchmark viscous behaviours that the

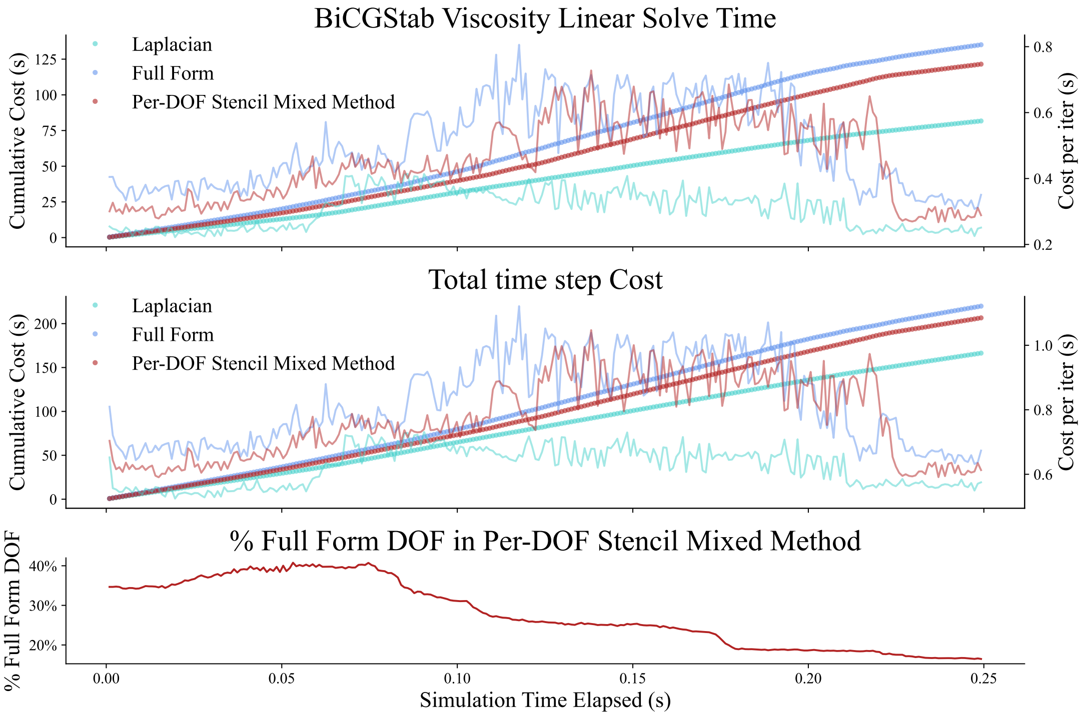


Figure 4.5: BiCGStab Solver (ViennaCL) comparison for Mixed Per-DOF Stencil Method for $N = 150$. This example follows the same parameters and simulation shown in Figure 4.4. Similar plots arise for when comparing computational costs for a fixed preconditioned BiCGStab solver, or a fixed preconditioned CG solver for the Free Surface Mixed method.

original unaltered Full Form Viscosity gives in this scene are:

1. The disk rotating down the solid wall
2. The beam bending
3. The middle column buckling side to side

These three benchmark viscous behaviours can be seen in Figure 4.4 in Frame 180 for the Full Form Viscosity Benchmark. As expected, the Laplacian Viscosity Benchmark fails to exhibit viscous rotating, bending and buckling due to incorrect boundary conditions.

Since our mixed method is a combination of the two, we use them as benchmarks in terms of visual qualitative behaviour and speed. We present visual comparisons of the full fluid solver with altered viscosity steps against the benchmarks in Figure 4.4 for chosen parameters which give 35% Laplacian DOFs in the interior of the liquid for the mixed methods. Both mixed methods use the same Square-radius Neighbor Liquid Content algorithm to construct identical inner discretization boundaries. The corresponding cumulative computational costs of the Mixed Per-DOF Stencil Method and the benchmarks, for a *fixed* BiCGStab solver, are shown in Figure 4.5 demonstrating that both the viscosity solve and overall simulation costs decrease relative to the Full Form Benchmark for a given fixed iterative solver. Similarly, speed-ups of the Free Surface Mixed Method occur relative to the Full Form Benchmark for a fixed preconditioned CG solver.

The 35% Full Form DOFs in the mixed methods of Figure 4.4 is relatively low for this scene, as there are only small amounts of Laplacian DOFs in the interior of the fluid for this simulation. Laplacian DOFs only begin to disperse themselves throughout the lower-middle portion of the column later as the simulation progresses. We also examine the same scene but with all blobs of fluid involving mostly Laplacian inner DOFs, 68% specifically. Figure 3.2 depicts a visual representation of the initial frame's DOF count that are Full Form vs Laplacian Form. In this example, the discretization boundary is much closer to the liquid-air surface and is pushing the mixed methods towards the limit in terms of Laplacian DOF's for this N value. The resultant mixed viscosity matrices are therefore sparser than the previous example. Simulation results for a single mid-simulation frame are shown in Figure 4.6, and we isolate the 3 individual blobs of fluid to easily compare the three benchmark viscous behaviours without the blobs of fluid interacting with one another. Computational costs for all 3 blobs of fluid simulated *together* are outlined in Table 4.2. From Figure 4.6 it is clear that the Per-DOF Mixed Method qualitatively exhibits two of the three viscous benchmark behaviours. It gives convincing rotation of the disk and bending of the beam, but it does not exhibit viscous buckling of the column and the column evolves similar to the Laplacian benchmark. We do not show the Free Surface Mixed method in Figure 4.6 for this example with more Laplacian DOFs as it exhibited none of the three viscous benchmarks. Table 4.2 shows overall speedups for both mixed methods relative to the Full Form benchmark for the viscosity linear solve, with minimal speedup for the Per-DOF Mixed method.

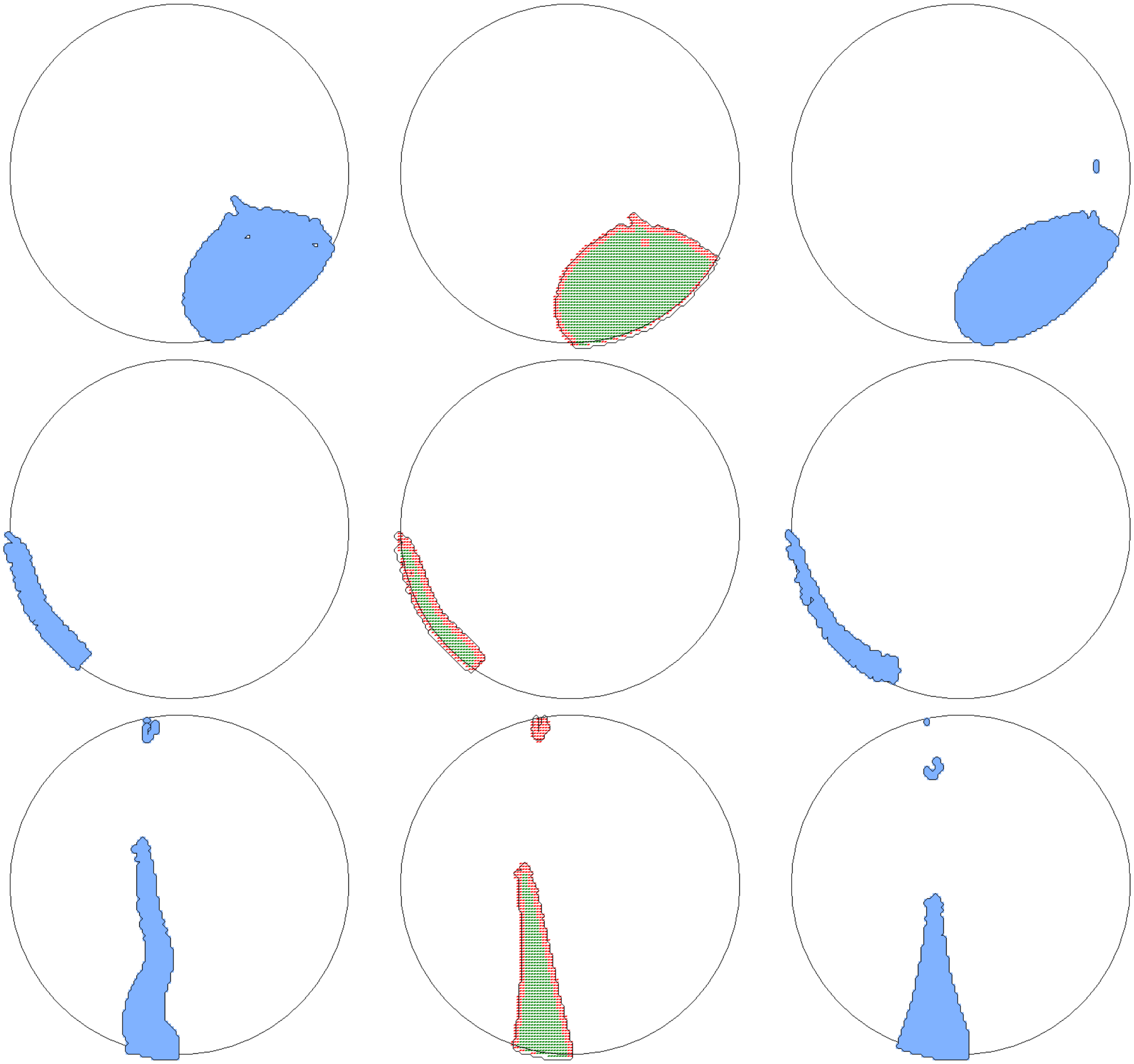


Figure 4.6: Scene 1 with a sparser mixed example with 68% Laplacian DOFs in the per-DOF Mixed Method vs the benchmarks for $N = 150$, all at frame 180. The average $\mathrm{NNZ}(A) = 91527$, 74496, 68292 respectively for the Full Form Benchmark, Per-DOF Stencil Mixed, and Laplacian Benchmark.

Discretization	Solver	Preconditioner	Package	Viscosity Solve (s)
Laplacian	CG	ICHOL	Built-in	**0.161**
Laplacian	CG	Block-ILU0	ViennaCL	0.177
Laplacian	BiCGStab	Block-ILU0	ViennaCL	0.196
Laplacian	CG	none	ViennaCL	0.221
Laplacian	BiCGStab	none	ViennaCL	0.315
Free Surface	CG	ICHOL	Built-in	**0.163**
Free Surface	CG	Block-ILU0	ViennaCL	0.168
Per-DOF Stencil	BiCGStab	Block-ILU0	ViennaCL	**0.249**
Per-DOF Stencil	BiCGStab	none	ViennaCL	0.419
Full Form	CG	Block-ILU0	ViennaCL	**0.250**
Full Form	BiCGStab	Block-ILU0	ViennaCL	0.271
Full Form	CG	ICHOL	Built-in	0.368
Full Form	CG	none	ViennaCL	0.382
Full Form	BiCGStab	none	ViennaCL	0.531

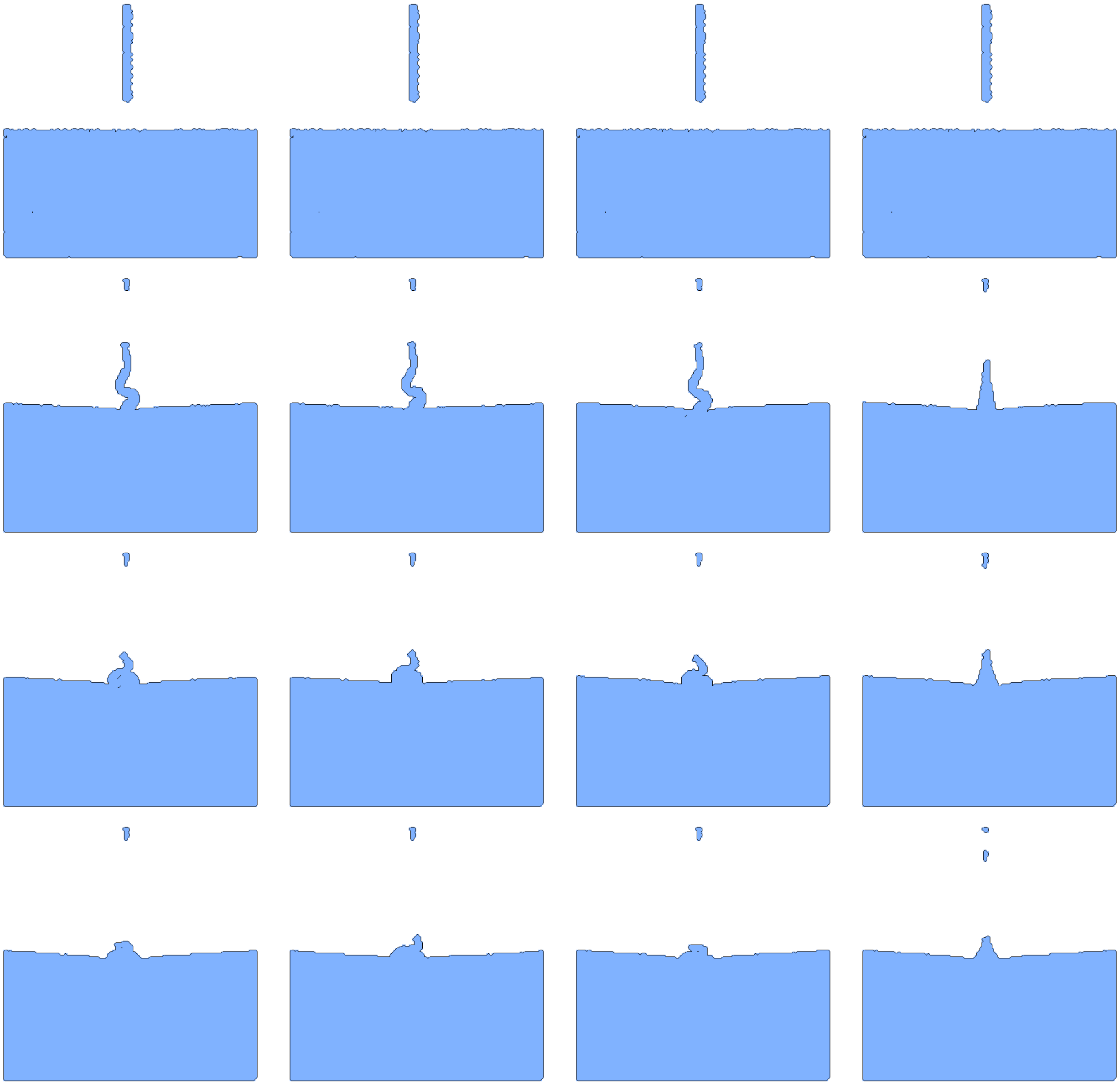
Table 4.2: Scene 1 average viscosity solve computational costs for the sparser example with 68% Laplacian DOFs in the mixed methods. 310 time steps are used and $N = 150$.

4.2.2 Scene 2: Rectangular Boundary with Column and Pile

We alter the same fluid solver code base to create a scene involving a column of liquid at rest falling into a large vast rectangular pile of fluid, all within an outer rectangular solid boundary (Figure 4.7). The large pile of liquid enables our mixed methods to take large advantage of the interior of the fluid, with 90.5% of the DOFs on average being cheaper Laplacian DOFs in this scene for the chosen parameters. See Figure 3.2 for the dispersion of the Laplacian DOFs vs the Full Form DOFs at the initial condition for the mixed methods. As time evolves, the Laplacian DOFS enter into the falling column after it makes contact with the rectangular pile of fluid. Since we only have a column falling from rest, the single benchmark behaviour we seek in this scene is viscous buckling of the column. Looking at frame 150 of Figure 4.7 (second row), we see convincing viscous buckling of both mixed methods which closely mimics the Full Form Benchmark. Corresponding average computational costs for the viscosity solve are outlined in Table 4.3. Overall, viscosity solves for the mixed methods are faster than the Full Form benchmarks. The Per-DOF Stencil mixed method with BiCGStab has a marginal 2.0% average viscous solve speed-up, while the Free Surface mixed method with CG shows a 20.9% average viscous solve speed-up.

Discretization	Solver	Preconditioner	Package	Viscosity Solve (s)	Time Step Speed (s)
Laplacian	CG	ICHOL	Built-in	**0.084**	0.364
Laplacian	BiCGStab	none	ViennaCL	0.203	0.636
Laplacian	GMRES	none	ViennaCL	0.998	1.424
Laplacian	CG	ICHOL	Eigen	0.922	1.356
Laplacian	BiCGStab	ILUT	Eigen	1.641	2.068
Free Surface	CG	ICHOL	Built-in	**0.272**	0.585
Per-DOF Stencil	BiCGStab	none	ViennaCL	**0.337**	0.785
Free Surface	CG	ICHOL	Eigen	1.605	2.074
Per-DOF Stencil	GMRES	none	ViennaCL	1.933	2.379
Per-DOF Stencil	BiCGStab	ILUT	Eigen	1.963	2.406
Full Form	CG	ICHOL	Built-in	**0.344**	0.634
Full Form	BiCGStab	none	ViennaCL	0.434	0.940
Full Form	GMRES	none	ViennaCL	2.295	2.800
Full Form	CG	ICHOL	Eigen	2.348	2.883
Full Form	BiCGStab	ILUT	Eigen	3.623	4.124

Table 4.3: Scene 2 average viscous solve and full simulation step costs corresponding to Figure 4.7. 90.5% of the DOFs on average are Laplacian DOFs. 510 time steps are used and $N = 180$.



Full Form Benchmark Per-DOF Stencil Free Surface Laplacian Benchmark

Figure 4.7: Simulation comparison of a viscous stream of liquid falling from rest into a square-boundary for $N = 180$ at frames [1, 150, 200, 300]. An average of 90.5% of the active DOF are Laplacian DOF for both the Per-DOF Stencil and Free Surface mixed methods, while qualitatively retaining the viscous buckling behaviour of the Full Form Benchmark. Computational costs are outlined in Table 4.3

Chapter 5

Conclusion

In this research project, we developed novel mixed viscosity discretizations to address efficiency issues regarding Eulerian grid based viscosity integration via a drop-in replacement of the operator-split viscous portion of the incompressible Navier-Stokes fluid equations. Specifically, we combined the computer graphics industry-accepted Full Form Viscosity model with a cheaper reduced Laplacian Viscosity model by employing the Full Form near the liquid-air surface, while employing the Laplacian Form in the interior of the liquid. This increases matrix sparsity as the Laplacian Form and Full Form have respectively five-point uncoupled and nine-point velocity-component-coupled stencils in 2D.

We introduced three novel mixed discretization coupling approaches: the Per-DOF Stencil Selection Method, the pseudo inner-liquid Free Surface Method, and the Smoothing-α approach between the energy formulation point of view of the PDEs. The latter two methods retain SPD matrices, while the first loses symmetry. We also propose an algorithm to determine the inner-liquid discretization splitting boundary location that considers the local neighborhood of the surrounding velocity DOFs. These mixed methods have relatively low overhead cost to implement, with the goal of speeding up the viscous linear solve.

We proposed a mixed spatial discretization test in 2D for the viscosity step alone with three test case functions and two mixed domain scenarios to evaluate each mixed method. The Per-DOF Stencil method obtained spatial convergence to first order of the L_∞ error for all tests, while the two symmetric mixed methods converged for only one of the six tests, validating our intuition that these two symmetric mixed methods unfortunately enforce pseudo inner-liquid boundaries that indeed introduce a non-negligible error in the resultant velocity field.

We then incorporated and evaluated the Per-DOF Stencil and Free Surface Mixed methods as drop-in viscosity solve replacements to a standard 2D fluid simulator. In scene 1 we saw in Figure 4.4 with 35% of the DOFs being Laplacian that all three benchmark viscous behaviours are retained for the Per-DOF Stencil mixed method and two of the three are retained for the Free Surface Mixed method (no rotation of circle). We show that for a fixed solver, speedups occur. We then investigate the same scene but with the mixed methods having 68% Laplacian DOFs, increasing matrix sparsity compared to the previous mixed method, and we see that qualitatively still two of the three proper viscous behaviour benchmarks are achieved for the Per-DOF mixed method. the Free Surface Mixed method did not achieve any of the benchmark viscous behaviours for this sparser scenario. Both mixed methods achieve *overall* viscous solve speedups to the Full Form benchmark's fastest preconditioned CG solve in this sparser scenario, with marginal increase for the Per-DOF Stencil Mixed method. A second scene is examined, where a column of liquid falls from rest into a very large rectangular pile of liquid in which 90.5% of the DOFs are Laplacian Form DOFs and we see a marginal 2.0% viscous solve speed-up from the Per-DOF Mixed Method with BiCGStab, while the Free Surface Mixed Method with preconditioned CG shows a 20.9% viscous solve speed-up. Both mixed methods show the correct benchmark viscous behaviour for this second scene.

Mixed discretizations show promising potential to address the viscosity speed issues of computer graphics. An important avenue of future work would be to develop a discretization resulting in a SPD matrix which can consistently exhibit all three viscous behaviour benchmarks, while being cheaper than the Full Form benchmark. Minimizing the Laplacian energy functional $E_0[\vec{u}]$ in Equation 2.17 with an additional energy term involving the surface integral $\iint_{\partial\Omega} \mu u^T (\nabla u)^T n \; dA$ to ensure the natural boundary condition is the correct operator-split zero-traction boundary condition 2.9 may work. Goldade et al. [13] show that acceleration via octree/quadtree spatial adaptivity within the inner liquid is possible for both discretizations we combined in this project, and thus another future work avenue would be to combine their dimension reduction technique with our matrix sparsity boosting technique. Alternatively, future work could be done to employ a mixed discretization between the pressure-velocity coupled unsteady Stokes viscosity solver presented by Larionov et al. [16] and the simpler Laplacian Viscosity. Starting from and adjusting this higher-accuracy coupled solver may better enforce both incompressibility and the full zero traction boundary condition, which may exhibit better qualitative results from analogously employing a mixed discretization method. Other potential avenues of exploration are preconditioners for mixed-discretization methods such as the ones presented here.

References

[1] Mridul Aanjaneya, Ming Gao, Haixiang Liu, Christopher Batty, and Eftychios Sifakis. Power diagrams and sparse paged grids for high resolution adaptive liquids. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[2] Mridul Aanjaneya, Chengguizi Han, Ryan Goldade, and Christopher Batty. An efficient geometric multigrid solver for viscous liquids. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1–21, 2019.

[3] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics (TOG)*, 26(3):100–es, 2007.

[4] Christopher Batty and Robert Bridson. Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*, pages 219–228, July 2008.

[5] Christopher Batty and Ben Houston. A simple finite volume method for adaptive viscous liquids. In *Proceedings of the 2011 ACM SIGGRAPH/eurographics symposium on computer animation*, pages 111–118, 2011.

[6] Robert Bridson. *Fluid simulation for computer graphics.* CRC press, 2015.

[7] Mark Carlson, Peter J Mucha, R Brooks Van Horn III, and Greg Turk. Melting and flowing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167–174, 2002.

[8] Nuttapong Chentanez, Bryan E Feldman, François Labelle, James F O'Brien, and Jonathan R Shewchuk. Liquid simulation on lattice-based tetrahedral meshes. 2007.

[9] Henrik Fält and Douglas Roble. Fluids with extreme viscosity. In *ACM SIGGRAPH 2003 Sketches & Applications*, pages 1–1. 2003.

[10] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, 2001.

[11] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical models and image processing*, 58(5):471–483, 1996.

[12] Ryan Goldade, Mridul Aanjaneya, and Christopher Batty. Constraint bubbles and affine regions: reduced fluid models for efficient immersed bubbles and flexible spatial coarsening. *ACM Transactions on Graphics (TOG)*, 39(4):43–1, 2020.

[13] Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. An adaptive variational finite difference framework for efficient symmetric octree viscosity. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[14] Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

[15] Stam Jos. Stable fluids. *Proceedings of SIGGRAPH99, Computer Graphics Proceedings*, pages 121–128, 1999.

[16] Egor Larionov, Christopher Batty, and Robert Bridson. Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017.

[17] Frank Losasso, Frédéric Gibou, and Ron Fedkiw. Simulating water and smoke with an octree data structure. In *Acm siggraph 2004 papers*, pages 457–462. 2004.

[18] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *Computers & Graphics*, 13(3):305–309, 1989.

[19] Yen Ting Ng, Chohong Min, and Frédéric Gibou. An efficient fluid–solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228(23):8807–8829, 2009.

[20] Stanley Osher, Ronald Fedkiw, and K Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004.

[21] Jonathan Panuelos, Ryan Goldade, Eitan Grinspun, David Levin, and Christopher Batty. Polystokes: A polynomial model reduction method for viscous fluid simulation. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023.

[22] Stéphane Popinet. Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of computational physics*, 190(2):572–600, 2003.

[23] Doug Roble, Nafees bin Zafar, and Henrik Falt. Cartesian grid fluid simulation with irregular boundary voxels. In *ACM SIGGRAPH 2005 Sketches*, pages 138–es. 2005.

[24] Lin Shi and Yizhou Yu. Visual smoke simulation with adaptive octree refinement. In *Computer Graphics and Imaging*, pages 13–19, 2004.

[25] SideFX. Houdini, 2023.

[26] Demetri Terzopoulos, John Platt, and Kurt Fleischer. From goop to glop: Heating and melting deformable models. In *Proceedings Graphics Interface*, volume 2, pages 219–226, 1989.

[27] Yipeng, Wang. Viscous liquid animation with spatially adaptive grids. Master's thesis, 2018.

APPENDICES

Appendix A

Derivations

A.1 Minimizing $E_\alpha[\vec{u}]$

We discretize and store volume fractions in arrays V^u, V^v, V^p, $V^{\tau^{12}}$, to respectively denote volume fractions which surround u, v, p, and τ^{12}. $V^u_{i+\frac{1}{2},j}$, $V^v_{i,j+\frac{1}{2}}$, V^p_{ij}, $V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}$ surround $u_{i+\frac{1}{2},j}$, $v_{i,j+\frac{1}{2}}$, p_{ij}, and $\tau^{12}_{i+\frac{1}{2},j+\frac{1}{2}}$. See Figure 2.2 for control volumes and quantity locations.

Similarly to volume fractions we define arrays for α which are piecewise constant, attached to each term in the discrete summations, with $\alpha^u_{i+\frac{1}{2},j}$, $\alpha^v_{i,j+\frac{1}{2}}$, α^p_{ij}, $\alpha^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}$ corresponding to the summation terms attached to the respective volume fractions of the same indices.

Note in the below that for when all α values are one, we have equivalently minimized the Full Form Viscosity smooth energy $E_1[\vec{u}]$ of Equation 2.11. For when all α values are zero, we have equivalently minimized the Laplacian viscosity energy $E_0[\vec{u}]$ of Equation 2.17.

Along with the discrete sums in Equation 2.12, the last sums needed before minimizing are:

$$\iint_\Omega \mu \left\|\nabla \vec{u}\right\|_F^2 dA = \iint_\Omega \mu \left\| \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \right\|_F^2 dA = \iint_\Omega \mu \left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dA$$

$$\begin{aligned} &\approx \sum_{i,j\in\Omega} V^p_{ij}\mu_{ij} \left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}\right)^2 \Delta x^2 + \sum_{i,j\in\Omega} V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}}{\Delta x}\right)^2 \Delta x^2 \\ &+ \sum_{i,j\in\Omega} V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\mu_{i+\frac{1}{2},j+\frac{1}{2}} \left(\frac{v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}}{\Delta x}\right)^2 \Delta x^2 + \sum_{i,j\in\Omega} V^p_{ij}\mu_{ij} \left(\frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x}\right)^2 \Delta x^2 \end{aligned}$$

With Equation 2.12 and the above discrete sums on hand, we now evaluate

$$E_\alpha[\vec{u}] = \iiint_\Omega \left(\frac{\rho}{2\Delta t}\left\|\vec{u} - \vec{u}^{old}\right\|^2 + \alpha\left(\mu\left\|\frac{\nabla\vec{u} + (\nabla\vec{u})^T}{2}\right\|_F^2\right) + \frac{(1-\alpha)}{2}\left(\mu\left\|\nabla\vec{u}\right\|_F^2\right)\right) dV \tag{A.1}$$

which is clearly equivalent to $E_1[\vec{u}]$ in Equation 2.11 for when all α arrays are one, and equivalent to $E_0[\vec{u}]$ in Equation 2.17 for when all α arrays are zero.

Differentiating $E_\alpha[\vec{u}]$ with respect to $u_{i+\frac{1}{2},j}$, fixing viscosity to be a constant $\tilde{\mu}$, cancelling out $\frac{\Delta x^2}{\Delta x^2}$ and equating to zero we get

$$\begin{aligned} 0 =& \frac{1}{\Delta t} V^u_{i+\frac{1}{2},j}\rho_{i+\frac{1}{2},j}\left(u_{i+\frac{1}{2},j} - u^{old}_{i+\frac{1}{2},j}\right)\Delta x^2 \\ &+ 2\tilde{\mu}V^p_{ij}\alpha^p_{ij}\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right) - 2\tilde{\mu}V^p_{i+1,j}\alpha^p_{i+1j}\left(u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}\right) \\ &- \tilde{\mu}V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\alpha^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\left(u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j} + v_{i+1,j+\frac{1}{2}} - v_{i,j+\frac{1}{2}}\right) \\ &+ \tilde{\mu}V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\alpha^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\left(u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1} + v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}}\right) \\ &+ \tilde{\mu}V^p_{ij}(1-\alpha^p_{ij})\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right) - \tilde{\mu}V^p_{i+1,j}(1-\alpha^p_{i+1,j})\left(u_{i+\frac{3}{2},j} - u_{i+\frac{1}{2},j}\right) \\ &- \tilde{\mu}V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}(1-\alpha^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}})\left(u_{i+\frac{1}{2},j+1} - u_{i+\frac{1}{2},j}\right) + \tilde{\mu}V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}(1-\alpha^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}})\left(u_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j-1}\right) \end{aligned}$$

which reduces to

$$\begin{aligned}
V^{u}_{i+\frac{1}{2},j}\rho^{u}_{i+\frac{1}{2},j}u^{old}_{i+\frac{1}{2},j} = &\left(V^{u}_{i+\frac{1}{2},j}\rho^{u}_{i+\frac{1}{2},j} + (1+\alpha^{p}_{ij})\tilde{k}V^{p}_{ij} + (1+\alpha^{p}_{i+1,j})\tilde{k}V^{p}_{i+1,j} + \tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}} + \tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\right)u_{i+\frac{1}{2},j} \\
&- (1+\alpha^{p}_{ij})\tilde{k}V^{p}_{ij}u_{i-\frac{1}{2},j} - (1+\alpha^{p}_{i+1,j})\tilde{k}V^{p}_{i+1,j}u_{i+\frac{3}{2},j} \\
&- \tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}u_{i+\frac{1}{2},j+1} - \tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}u_{i+\frac{1}{2},j-1} \\
&+ \alpha^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j+\frac{1}{2}}\left(v_{i,j+\frac{1}{2}} - v_{i+1,j+\frac{1}{2}}\right) + \alpha^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\tilde{k}V^{\tau^{12}}_{i+\frac{1}{2},j-\frac{1}{2}}\left(v_{i+1,j-\frac{1}{2}} - v_{i,j-\frac{1}{2}}\right)
\end{aligned}$$

with $\tilde{k} = \frac{\Delta t\,\tilde{\mu}}{\Delta x^2}$. Degrees of freedom that have a linear equation with *all* α equal to one use the pure Full Form viscosity stencil Equation 2.13. Degrees of freedom that have a linear equation with *all* α equal to zero use the pure Laplacian stencil Equation 2.18.