A Higher-Order Upwind Method for Viscoelastic Flow

By

ANDREW JEFFREY TADAO NONAKA
B.S. (University of the Pacific) 2002
M.S. (University of California, Davis) 2004

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Engineering - Applied Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

_____

_____

_____

Committee in Charge

2007

A Higher-Order Upwind Method for Viscoelastic Flow

by

Andrew Jeffrey Tadao Nonaka

Andrew Jeffrey Tadao Nonaka

September 2007

Engineering - Applied Science

A Higher-Order Upwind Method for Viscoelastic Flow

Abstract

A conservative finite difference method designed to capture elastic wave propagation in viscoelastic fluids in two space dimensions is presented. The governing equations are the incompressible Navier-Stokes equations coupled to the Oldroyd-B constitutive equations for viscoelastic stress. The equations are cast into a hybrid conservation form to make use of a second-order upwind method to treat the hyperbolic part of the equations. The hyperbolic step also utilizes a new exact and efficient Riemann solver. A numerical stress splitting technique provides a well-posed discretization for the entire range of Newtonian and elastic fluids. Incompressibility is enforced through both a projection method and a special partitioning of variables which suppresses compressive waves in the hyperbolic step. An embedded boundary approach for irregular geometry is employed, in which regular Cartesian cells are cut into irregular control volumes, requiring special discretization stencils. The resulting method is second-order accurate in L^1 for smooth geometries for a range of Oldroyd-B fluids.

Contents

Glossary

Geometry Description:

$A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$	Face $\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d$
$A_{\mathbf{i},EB}$	EB face
$C_{\mathbf{i}}$	Cartesian cell $\mathbf{i}$
$\partial C_{\mathbf{i}}$	Edges of Cartesian cell $C_{\mathbf{i}}$
$\mathbf{e}_d$	d^{th} unit vector
h	Grid spacing
$\mathbf{i} = ij$	Location index
$\mathbf{n}_{\mathbf{i}}$	Normalized average outward normal of $\partial\Omega$ over $A_{\mathbf{i},EB}$
$V_{\mathbf{i}}$	Control volume $\mathbf{i}$
$\partial V_{\mathbf{i}}$	Edges of control volume $V_{\mathbf{i}}$
$\mathbf{x}_{\mathbf{i}\pm\mathbf{e}_d}$	Location of face centroid on face $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$
$\hat{\mathbf{x}}_{\mathbf{i}\pm\mathbf{e}_d}$	Normalized location of face centroid on face $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$
$\mathbf{x}_{\mathbf{i},EB}$	Location of face centroid on face $A_{\mathbf{i},EB}$
$\hat{\mathbf{x}}_{\mathbf{i},EB}$	Normalized location of face centroid on face $A_{\mathbf{i},EB}$
$\alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$	Area fraction of face $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$
$\alpha_{\mathbf{i},EB}$	Area fraction of EB face $A_{\mathbf{i},EB}$
$\kappa_{\mathbf{i}}$	Volume fraction of control volume $V_{\mathbf{i}}$
$\boldsymbol{\upsilon}$	Vector whose entries are all one
Ω	Solution domain
$\partial\Omega$	Solution domain boundary

Symbols:

CFL	CFL number
De	Deborah number
M	Acoustic Mach number
Ma	Elastic Mach number
Re	Reynolds number
We	Weissenberg number
a^2	Wave speed for numerical stress splitting
$\mathbf{A}$	Primitive coefficient matrix
b	Right-hand-side of an elliptic or parabolic operator
$\mathbf{b}$	Body force vector
$\mathbf{B}$	Finger tensor
c_d	Notation for the elastic shear wave speed, $\sqrt{(a^2 + \tau_{dd})/\rho}$
c_s	Speed of sound in a fluid
$\mathbf{C}$	Cauchy-Green tensor
D	Diffusion or Helmholtz equation constant
$\mathbf{D}$	Rate-of-strain tensor
E	Young's (elastic) modulus

$\mathcal{E}$	Error between computed solutions at different resolutions
f	Source term in diffusion equation
$\mathbf{f}$	Deformation gradient
$\mathbf{F}$	Tensor of fluxes
$\mathbf{g}$	Inverse deformation gradient
H	Heaviside step function
k	Iteration index for iterative matrix solver
$\mathbf{l}$	Left row eigenvector
L	Characteristic length
L^N	Error norm
$\mathbf{L}$	Matrix of left row eigenvectors
$\boldsymbol{\mathcal{L}}$	Matrix corresponding to an elliptic or parabolic operator
$\mathbf{M}$	The conserved variable $\mathbf{M} = \mathbf{g} \cdot (a^2\mathbf{I} + \boldsymbol{\tau}) \cdot \mathbf{g}^T$
n	Time step index
N_A	Number of active primitive variables
N_I	Number of inactive primitive variables
N_W	Number of primitive variables
N_U	Number of conserved variables
N_1	First normal stress difference
N_2	Second normal stress difference
p	Hydrostatic pressure
P^N	Order of convergence corresponding to L^N norm
$\mathbf{P}$	The projection operator
$\mathbf{Q}$	The projection remainder operator
$\mathbf{r}$	Right column eigenvector
$\mathbf{R}$	Matrix of right column eigenvectors
$\mathcal{R}$	Riemann solver
$\boldsymbol{\mathcal{R}}$	Rotation matrix
S	Entropy
S_i	Wave speed in Riemann solver
$\mathbf{S}_W$	Primitive source term vector
$\mathbf{S}_U$	Conserved source term vector
t	Time
Δt	Time step
$\mathbf{t}$	An arbitrary second-order tensor
T	Characteristic time
$\mathbf{T}$	Extra-stress tensor
$\mathbf{u}$	Velocity vector
U	Characteristic velocity
$\mathbf{U}$	Conserved variable vector
$\mathbf{v}$	An arbitrary vector
$\mathbf{W}$	Primitive variable vector
$\mathbf{x}$	Spatial (Eulerian) coordinate
$\mathbf{X}$	Material (Lagrangian) coordinate

α_d	Notation for $a^2 + \tau_{dd}$
β	Weighting coefficient for eigenvector decomposition
$\dot{\gamma}$	Shear rate
$\boldsymbol{\delta}$	Amount of conserved variable to be redistributed
ϵ	Elongational strain
ϵ_{PTT}	Parameter in the Phan-Thien Tanner model
ζ	Damping factor for cell-centered projection filter
η	Diffusion equation solver coefficients
λ	Relaxation time
Λ	Eigenvalue
$\Lambda_{\max}$	Eigenvalue of largest magnitude on the domain
$\boldsymbol{\Lambda}$	Set of eigenvalues
μ	Characteristic viscosity
μ_b	Bulk viscosity
μ_e	Elongational viscosity
μ_E	Uniaxial extensional viscosity
μ_p	Polymeric contribution to total viscosity
μ_s	Solvent contribution to total viscosity
ν	Kinematic viscosity
ξ	Machine precision
ρ	Density
$\boldsymbol{\sigma}$	Surface stress tensor
τ	Parameter in a^2 calculation
τ_e	Elongational stress
$\boldsymbol{\tau}$	Polymeric contribution to the extra-stress tensor
$\boldsymbol{\tau}_s$	Solvent contribution to the extra-stress tensor
ϕ	An arbitrary scalar
χ	Parameter used in a^2 calculation
χ_{G}	Parameter in the Giesekus model
χ_{PTT}	Parameter in the Phan-Thien Tanner model
Ψ	Wave strength in Riemann solver
ω	Weighting coefficient for iterative solver

List of Figures

List of Tables

Acknowledgments

I would like to thank my thesis advisor, Greg Miller, for encouraging and believing in me. Greg was always able to make me trust in my own abilities to work through any problem. I would also like to thank my Lawrence Livermore technical advisor, David Trebotich, for his support and guidance in the ways of computational viscoelasticity. A special thanks goes to Phil Colella, Dan Graves, and the Lawrence Berkeley Applied Numerical Algorithms Group for their software and numerical expertise. Funding for my journey through graduate school was graciously provided by the Lawrence Livermore Student Employee Graduate Research Fellowship.

I also wish to thank my friends and coaches at the Davis Aquatic and Walnut Creek Masters, for helping keep balance in life. Most importantly, I would like to thank my parents for helping me get to graduate school, and my wife, Marissa, for helping me get through graduate school. Marissa experienced the joys and pains of graduate school without actually being there, and is the only person happier about being "done" than myself.

Introduction

This work describes the development of a numerical algorithm for time-dependent viscoelastic fluids in two space dimensions. There is an ever-increasing need for accurate simulation tools for complex fluids found in many industrial and scientific applications, and viscoelasticity is an important property of many such systems. The chosen model for viscoelasticity is the incompressible Navier-Stokes equations coupled to the Oldroyd-B constitutive equations for viscoelastic stress. The Oldroyd-B equations are a relatively simple continuum model for viscoelasticity, yet numerical solutions of such fluids even in simple systems are very difficult to obtain. Hopefully the techniques developed in this work can be applied to more advanced constitutive models to more accurately predict real-world phenomena.

This algorithm uses a conservative finite difference (or finite volume) approach, where the solution domain is divided into rectangular cells. The state variables, including velocity, stress, and pressure, are defined at the geometric centers of each cell, and represent the average value in the cell. An embedded boundary approach is used to represent irregular geometry, in which rectangular cells are cut into irregular control volumes, requiring special discretization stencils. The basic time-stepping strategy is to use an upwind procedure to compute intercell fluxes and advance the solution in time conservatively. A projection method is used to enforce incompressibility and the diffusive terms are treated using an advanced semi-implicit discretization. The resulting method is second-order accurate in L^1 for a range of Oldroyd-B fluids.

This thesis is divided into five chapters. Chapter 1 reviews relevant concepts in fluid mechanics. Chapter 2 gives an introduction to viscoelasticity, including examples of real-world viscoelastic phenomena and a mathematical analysis of the Oldroyd-B model. In Chapter 3, a history of computational viscoelasticity is presented along with an introduction to upwind and projection methods. Chapter 4

provides the details of the numerical method. Finally, in Chapter 5, results are presented for several Oldroyd-B fluids.

1 Review of Fluid Mechanics

1.1 Notation and Conventions

A variable in bold refers to a tensor quantity, which may be first-order (a vector) or second-order (a matrix). The order of a tensor will be clear in context. The subscripts 0 and 1 refer to the corresponding x and y components of a quantity. For example, x_0 refers to the x-direction, u_1 refers to the y-component of velocity, and σ_{01} refers to the xy-component of the surface stress tensor.

This thesis uses the standard definitions of the dot (or scalar) product and cross product used in graduate level fluid mechanics textbooks [12, 30]. Note that representing a first-order tensor as a column vector and a second-order tensor as a matrix is useful in that a dot product involving one or more second-order tensors is equivalent to standard matrix multiplication. The lack of a dot indicates the dyadic product, which is usually defined as a formal product of two vectors, but here has been extended to tensors of arbitrary order. For example, if $\mathbf{v}$ is a vector and $\mathbf{t}$ is a second-order tensor:

$$(\mathbf{vt})_{ijk} = v_i t_{jk}, \qquad (\mathbf{vv})_{ij} = v_i v_j. \tag{1.1}$$

One change will be made to the conventions in [12, 30]. Continuum mechanics and applied mathematics conventions differ for the compact notation used in the gradient operating on a vector. Here the following convention is used:

$$(\nabla \mathbf{v})_{ij} = \frac{\partial v_i}{\partial x_j} = \begin{bmatrix} \frac{\partial v_0}{\partial x_0} & \frac{\partial v_0}{\partial x_1} \\ \frac{\partial v_1}{\partial x_0} & \frac{\partial v_1}{\partial x_1} \end{bmatrix}. \tag{1.2}$$

A special derivative which is often used in fluid mechanics is the material derivative:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla, \tag{1.3}$$

which describes the rate of change of a quantity as seen by an observer moving at the fluid velocity, $\mathbf{u}$. This is in contrast to the Eulerian derivative, $\partial/\partial t$, which measures the rate of change of a quantity at a fixed point in space.

1.2 The Incompressible Navier-Stokes Equations

All fluids are governed by conservation of mass, also known as the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.4}$$

where ρ is the fluid density. This equation can be equivalently expressed using the material derivative:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \mathbf{u}) = 0. \tag{1.5}$$

All fluids are compressible to a certain extent, as can by quantified by examining the speed at which (acoustic) sound waves propagate in the fluid:

$$c_s = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_S}, \tag{1.6}$$

where p is the hydrostatic pressure and S is the entropy. For water at room temperature and 1.0 [atm] pressure, the speed of sound is $c_s = 1.5 \times 10^3$ [m/s]. Landau and Lifshitz [47] describe the two conditions for which a fluid can be considered as incompressible. First, the characteristic fluid velocity, U, must be much smaller than the speed of sound, i.e., $\mathrm{M} = U/c_s \ll 1$, where M is the acoustic Mach number. Second, the time required for a sound wave to travel the characteristic length of observation, L, must be small compared to the characteristic time of fluid observation, T, i.e., $L/c_s \ll T$. In this thesis, these two assumptions are considered valid, and the fluid will be modeled as incompressible.

Lai [46] performs an asymptotic expansion in M for the pressure and density, and shows that $D\rho/Dt = \mathcal{O}(\mathrm{M}^2)$ and $\nabla \cdot \mathbf{u} = \mathcal{O}(\mathrm{M}^2)$. Since $\mathrm{M} \ll 1$, any density

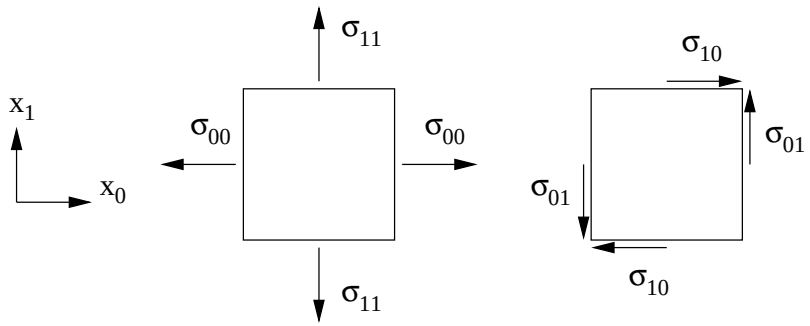

Figure 1.1: (Left) Normal stresses and (Right) shear stresses.

field which is initially constant in space is assumed to remain constant. Also, the following divergence constraint for incompressible fluids is revealed:

$$\nabla \cdot \mathbf{u} = 0. \tag{1.7}$$

All fluids are governed by conservation of momentum [30]:

$$\rho \frac{D\mathbf{u}}{Dt} = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma}. \tag{1.8}$$

Here, the body forces, $\mathbf{b}$, which may include, for example, the force due to gravity or a magnetic force on a fluid with net charge, are assumed to be zero. The surface stress tensor, $\boldsymbol{\sigma}$, is defined such that σ_{ij} is the force per unit area on a fluid element on a plane perpendicular to the i axis, acting in the j direction [30]. The diagonal components of $\boldsymbol{\sigma}$ are the normal stresses, while the off-diagonal components are the shear stresses, as illustrated in Figure 1.1. As a consequence of conservation of angular momentum in the absence of body torques, $\boldsymbol{\sigma}$ is symmetric [30].

For all fluids, $\boldsymbol{\sigma}$ consists of a contribution due to the hydrostatic pressure and an extra-stress term, $\mathbf{T}$:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{T}, \tag{1.9}$$

where $\mathbf{I}$ is the identity tensor. A fluid is said to be Newtonian if the extra-stress

tensor is of the form:

$$\mathbf{T} = \mu_s \underbrace{[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]}_{2\mathbf{D}} + \left(\mu_b - \frac{2}{3}\mu_s\right)(\nabla \cdot \mathbf{u})\mathbf{I}, \tag{1.10}$$

where μ_s is the viscosity of the fluid (or solvent), μ_b is the bulk viscosity, and $\mathbf{D} = [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]/2$ is the rate-of-strain tensor, which is a measure of fluid deformation given by the symmetric part of $\nabla \mathbf{u}$. Water is considered a very close approximation of a Newtonian fluid. A fluid is said to be non-Newtonian if the extra-stress tensor cannot be expressed by (1.10). The extra-stress may be given by a much more complicated constitutive relation or the change may be as simple as a viscosity which is a function of the velocity.

Applying the divergence constraint (1.7) to the extra-stress for a Newtonian fluid (1.10) leads to an extra-stress that is governed only by the viscous stress, $\boldsymbol{\tau}_s$, so that the surface stress for an incompressible Newtonian fluid is:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \underbrace{\mu_s \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right]}_{\boldsymbol{\tau}_s}. \tag{1.11}$$

Substituting (1.11) into (1.8) gives:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \left[\nabla(\nabla \cdot \mathbf{u}) + \Delta \mathbf{u}\right], \tag{1.12}$$

where $\nu = \mu_s/\rho$ is the kinematic viscosity and Δ is the Laplacian operator. Applying the divergence constraint (1.7) reduces this equation to the well-known evolutionary part of the incompressible Navier-Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u}. \tag{1.13}$$

A useful dimensionless number which can be found by non-dimensionalizing

(1.13) is the Reynolds number, which relates the relative strength of inertial forces to viscous forces in a fluid, given by:

$$\mathrm{Re} = \frac{\rho U L}{\mu}, \tag{1.14}$$

where μ is the characteristic viscosity, which in this case is equal to μ_s. A large Re indicates that the flow is dominated by inertial forces and a small Re indicates the flow is dominated by viscous forces.

1.3 Kinematics of Deformation

There are two different coordinate systems that can be used to analyze problems in mechanics. Using an Eulerian, or spatial coordinate system, "$\mathbf{x}$", the coordinates are fixed in space, and the properties of the fluid at these fixed points are tracked. Using a Lagrangian, or material coordinate system, "$\mathbf{X}$", the coordinates are attached to the fluid, and the properties of each particular fluid element are tracked. Mathematically, $\mathbf{x}$ is defined as the position occupied at time t by the particle which occupied position $\mathbf{X}$ in the reference configuration at $t = 0$ [51], such that:

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t), \qquad \mathbf{x}(\mathbf{X}, 0) = \mathbf{X}. \tag{1.15}$$

The fluid velocity is found as follows:

$$\mathbf{u}(\mathbf{x}(\mathbf{X}, t), t) = \left.\frac{D\mathbf{x}}{Dt}\right|_{\mathbf{X}} = \frac{\partial \mathbf{x}}{\partial t}. \tag{1.16}$$

The deformation gradient is a second-order tensor which relates $\mathbf{x}$ to $\mathbf{X}$ as follows:

$$\mathbf{f} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad \rightarrow \quad f_{ij} = \frac{\partial x_i}{\partial X_j}. \tag{1.17}$$

The inverse deformation gradient is defined as the inverse of $\mathbf{f}$:

$$\mathbf{g} = \mathbf{f}^{-1} = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \quad \rightarrow \quad g_{ij} = (\mathbf{f}^{-1})_{ij} = \frac{\partial X_i}{\partial x_j}. \tag{1.18}$$

Physically, $\mathbf{g}$ and $\mathbf{f}$ describe the deformation of a box whose corners are attached to the fluid. The physical meaning of each particular component of $\mathbf{g}$ and $\mathbf{f}$ can be easily seen by examining their evolution equations. The evolution equations for $\mathbf{g}$ can be derived in the following manner, using index notation and the material derivative:

$$\begin{aligned} (\nabla \mathbf{u})_{ij} &= \frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial X_k}\frac{\partial X_k}{\partial x_j} = \frac{\partial}{\partial X_k}\left(\frac{Dx_i}{Dt}\right)\frac{\partial X_k}{\partial x_j} = \frac{D}{Dt}\left(\frac{\partial x_i}{\partial X_k}\right)\frac{\partial X_k}{\partial x_j} \\ &= \left(\frac{D\mathbf{f}}{Dt}\cdot\mathbf{g}\right)_{ij}. \end{aligned} \tag{1.19}$$

Using the identity $\mathbf{f}\cdot\mathbf{g} = \mathbf{I}$ gives:

$$\frac{D(\mathbf{f}\cdot\mathbf{g})}{Dt} = \mathbf{f}\cdot\frac{D\mathbf{g}}{Dt} + \frac{D\mathbf{f}}{Dt}\cdot\mathbf{g} = 0. \tag{1.20}$$

Substituting (1.19) into (1.20) gives:

$$\mathbf{f}\cdot\frac{D\mathbf{g}}{Dt} + \nabla\mathbf{u} = 0. \tag{1.21}$$

Pre-multiply this equation by $\mathbf{g}$ and expand the material derivative to give the evolution equation:

$$\frac{\partial \mathbf{g}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{g} + \mathbf{g}\cdot(\nabla\mathbf{u}) = 0. \tag{1.22}$$

Similarly, an evolution equation for $\mathbf{f}$ can be obtained by substituting (1.20) into (1.21), post-multiplying by $\mathbf{f}$ and expanding the material derivative:

$$\frac{\partial \mathbf{f}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{f} - (\nabla\mathbf{u})\cdot\mathbf{f} = 0. \tag{1.23}$$

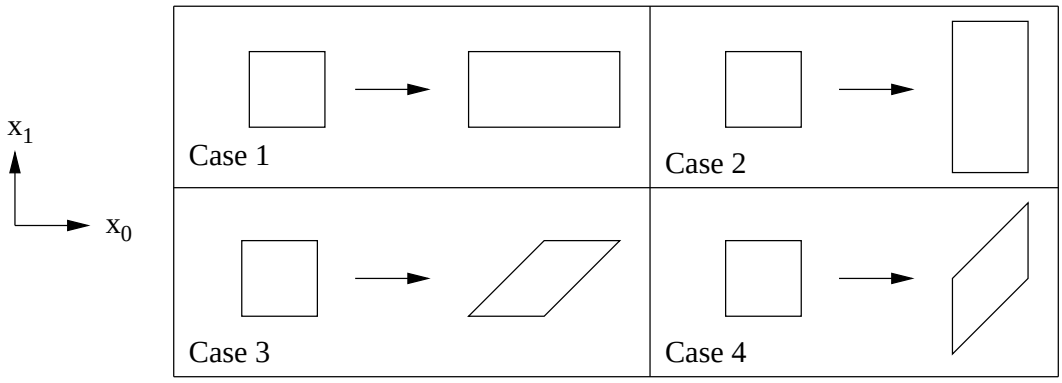

Figure 1.2: (Top) Elongational deformation and (Bottom) shear deformation.

To help comprehend the physical meaning of $\mathbf{f}$ and $\mathbf{g}$, first assume a velocity profile exists that is initially zero everywhere. This flow experiences no deformation and it is natural to set $\mathbf{f} = \mathbf{g} = \mathbf{I}$, reducing (1.22) and (1.23) to:

$$\frac{\partial \mathbf{g}}{\partial t} = -\nabla \mathbf{u}, \qquad \frac{\partial \mathbf{f}}{\partial t} = \nabla \mathbf{u}. \tag{1.24}$$

Now, "attach" a square box to this fluid and imagine the flow field instantaneously changes to one of the following 4 cases illustrated in Figure 1.2. These cases do not necessarily describe any real system, but rather are used for instructive purposes.

- Case 1, $\partial u_0/\partial x_0 > 0$, corresponds to a decrease in g_{00} and an increase in F_{00}.
- Case 2, $\partial u_1/\partial x_1 > 0$, corresponds to a decrease in g_{11} and an increase in F_{11}.
- Case 3, $\partial u_0/\partial x_1 > 0$, corresponds to a decrease in g_{01} and an increase in F_{01}.
- Case 4, $\partial u_1/\partial x_0 > 0$, corresponds to a decrease in g_{10} and an increase in F_{10}.

A simplified way of viewing these results is that the diagonal components of $\mathbf{g}$ and $\mathbf{f}$ are a measure of elongational deformation, whereas the off-diagonal components of $\mathbf{g}$ and $\mathbf{f}$ are a measure of shear deformation. Another property of the deformation gradients is that the determinant of $\mathbf{f}$ is proportional to the area bounded by the

box, A_{box}, and the determinant of $\mathbf{g}$ is inversely proportional to the area of the box, such that:

$$\det \mathbf{f} = \frac{1}{\det \mathbf{g}} \propto A_{\text{box}}. \tag{1.25}$$

For an incompressible fluid, A_{box} is constant in time. For the simple cases illustrated above, $\det \mathbf{f}$ and $\det \mathbf{g}$ remain constant under the shear deformation, whereas $\det \mathbf{f}$ increases and $\det \mathbf{g}$ decreases under elongation, which is consistent with (1.25).

2 Viscoelasticity

2.1 A Simple Viscoelastic Model

Viscoelastic fluids are a class of non-Newtonian fluids characterized by the presence of memory. Such a fluid combines elements of a Newtonian fluid and an elastic solid. Mathematically, memory can represented as an extra-stress term which is a function of the current and past velocity and stress, rather than just the current velocity as in the Newtonian case. A Maxwell element is a simple one-dimensional viscoelastic model, consisting of a spring and dashpot in series [50, 70], as illustrated in Figure 2.1. A spring exhibits ideal elastic behavior, that is, when subjected to an elongational stress, τ_e, it yields an elongational strain proportional to the stress, where strain is a dimensionless quantity defined as the normalized change in length, i.e., $\epsilon = (l - l_0)/l_0$. The constant of proportionality is Young's (elastic) modulus, E, yielding the constitutive relation:

$$\tau_e = E\epsilon. \tag{2.1}$$

A dashpot exhibits ideal viscous Newtonian behavior, that is, when subjected to an elongational stress, it undergoes a constant increase in elongational strain rate with respect to time. The constant of proportionality is the elongational viscosity, μ_e, in the constitutive relation:

$$\tau_e = \mu_e \frac{\partial \epsilon}{\partial t}. \tag{2.2}$$

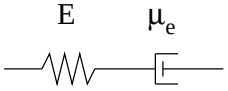

Figure 2.1: A Maxwell element consists of a spring and dashpot in series.

The stress and strain in a Maxwell element are related by [50, 70]:

$$\frac{\partial \tau_e}{\partial t} = \frac{\mu_e}{\lambda}\frac{\partial \epsilon}{\partial t} - \frac{1}{\lambda}\tau_e, \tag{2.3}$$

where $\lambda = \mu_e/E$ is the relaxation time. In the elastic limit ($\lambda \to \infty$, $E = \mu_e/\lambda$ finite), (2.3) reduces to the time derivative of (2.1). In the Newtonian limit ($\lambda \to 0$), (2.3) reduces to (2.2).

If a Maxwell element is subjected to a stress relaxation test in which the Maxwell element is stretched and held at a constant length, then $\partial\epsilon/\partial t = 0$ and τ_e represents the stress required to hold the Maxwell element at that constant length. The solution of equation (2.3) under these conditions is:

$$\tau_e(t) = \tau_e(0)e^{-t/\lambda}. \tag{2.4}$$

In the case of a Maxwell element, the relaxation time is a measure of how long it takes for the stress to relax to e^{-1} of the initial value in a stress relaxation test. From the above analysis, in the Newtonian limit, the stress relaxes instantly. In the elastic limit, the stress remains constant. Viscoelastic fluids are said to give an elastic response (no stress relaxation) over short time periods, and a Newtonian response (total stress relaxation) over long time periods. This particular phenomenon can be seen in Silly Putty®, which bounces like a rubber ball when tossed against a hard surface, yet deforms continuously when laid on a solid surface.

The relaxation time alone is not enough to fully characterize the elasticity of the fluid, since the characteristic velocity and length scales must be taken into account as well. The Weissenberg number is a dimensionless measure of the elasticity of a fluid, given by:

$$\mathrm{We} = \frac{\lambda U}{L}. \tag{2.5}$$

Similar to the Weissenberg number is the Deborah number. Rosen [70] defines the

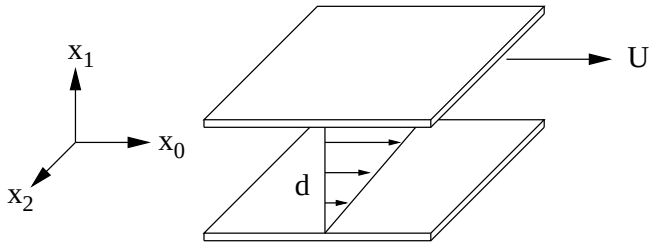

Figure 2.2: Steady shear flow between two parallel plates.

Deborah number as De = λ/T, or the ratio of the relaxation time to the characteristic time of observation, which is a much more qualitative statement than the previous definition of Weissenberg number. Marchal and Crochet [52] define the Deborah number as De = $\lambda\dot{\gamma}$, where $\dot{\gamma}$ is the shear rate on the wall for fully developed flow in a contraction channel. Phillips and Williams [65] point out that, for the case of a 2:1 planar contraction, this definition of the Deborah number differs from the Weissenberg number by a factor of three, such that De = 3We.

2.2 Examples of Viscoelastic Behavior

Viscoelastic fluids are found everywhere in daily life, and their behavior can be remarkably different than Newtonian fluids for even simple flows. Shear thinning and thickening, where the viscosity decreases or increases as a function of the shear rate, are two easily observable viscoelastic phenomena. Paint exhibits shear thinning such that paint applied to a wall may not flow unless a shear stress is applied with a brush. A corn starch mixture exhibits shear thickening such that when stirred, the solution becomes thicker. Other examples of viscoelastic behavior include non-zero normal stress differences in shear flow and a complex extensional viscosity function in elongational flow, both leading to surprising behavior which has profound implications in the mixing and transport of viscoelastic fluids.

Consider the steady shear problem, where in three dimensions two infinitely large parallel plates are separated by a distance d with the origin on the fixed lower

plate, and the upper plate moving with a constant velocity $u_0 = U$, as illustrated in Figure 2.2. For a viscous Newtonian fluid, the steady-state velocity profile is given as:

$$u_0(x_1) = \frac{U}{d} x_1 = \dot{\gamma} x_1, \tag{2.6}$$

where $\dot{\gamma} = U/d$ is the shear rate. The first and second normal stress differences are defined as:

$$N_1(\dot{\gamma}) = \sigma_{00} - \sigma_{11}, \qquad N_2(\dot{\gamma}) = \sigma_{11} - \sigma_{22}. \tag{2.7}$$

At any point in the fluid, the pressure is locally constant, so the normal stress differences can be determined by analyzing only the extra-stress, $\mathbf{T}$. The equations for surface stress in a Newtonian fluid (1.11) indicate that the normal stress differences in shear flow are zero. However, for a viscoelastic fluid, this may not be the case. For many viscoelastic fluids in shear flow, N_1 has been observed to be positive and N_2 has been observed to be negative with a much smaller magnitude than N_1 [6, 50].

Non-zero normal stress differences in shear flow can be qualitatively explained by considering certain viscoelastic fluids as Newtonian solvents laden with polymeric or biological molecules. These molecules stretch under an applied stress, and relax toward their original configuration when the stress is removed. These particles, when subjected to shear flow, will stretch and align themselves with the streamlines. Their attempt to relax back toward their original configuration causes them to exert a non-isotropic stress back on the fluid, causing the normal stress differences.

A well-known consequence of non-zero normal stress differences which is discussed in virtually all textbooks concerning viscoelastic fluids is the Weissenberg effect [6, 50, 81]. When a rotating rod is placed into a Newtonian fluid and a certain viscoelastic fluids such as a suspension of polystyrene particles, the results can be drastically different as illustrated in Figure 2.3. For the Newtonian fluid, inertial effects cause the fluid level to rise along the outer rim of the glass. For certain viscoelastic fluids, the normal stress differences combine with boundary effects, causing

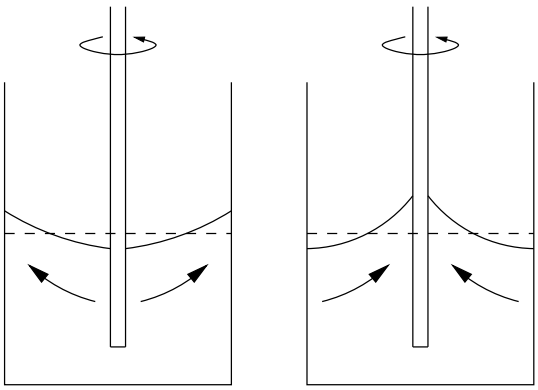

Figure 2.3: The Weissenberg effect. A rotating rod is inserted into a Newtonian fluid (Left) and a viscoelastic fluid (Right) yielding qualitatively different behavior.

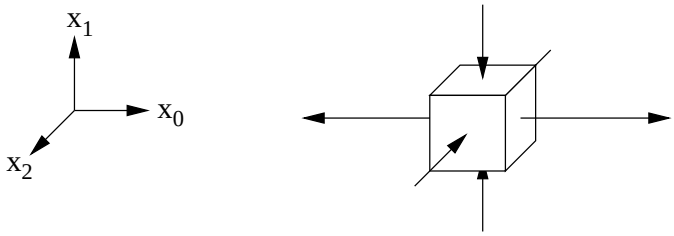

Figure 2.4: Uniaxial extension.

the fluid to stick to and climb up the rod.

As for extensional viscosity effects, consider a three-dimensional fluid element centered at the origin. This fluid element is said to undergo uniaxial extension if the fluid element deforms with the following velocity profile, as illustrated in Figure 2.4:

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} \dot{\epsilon} x_0 \\ -\frac{\dot{\epsilon}}{2} x_1 \\ -\frac{\dot{\epsilon}}{2} x_2 \end{bmatrix}, \tag{2.8}$$

where $\dot{\epsilon}$ is a constant strain rate. The (uniaxial) extensional viscosity is defined as:

$$\mu_E(\dot{\epsilon}) = \frac{\sigma_{00} - \sigma_{11}}{\dot{\epsilon}} = \frac{\sigma_{00} - \sigma_{22}}{\dot{\epsilon}}. \tag{2.9}$$

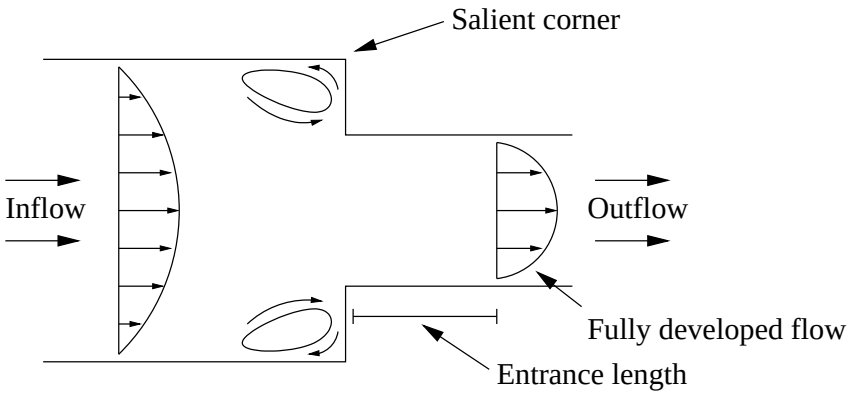

Figure 2.5: Recirculation zones and extended entry lengths have been observed in the flow of certain viscoelastic liquids through contraction channels.

The equations for surface stress in a Newtonian fluid (1.11) indicate that $\mu_E(\dot{\epsilon}) = 3\mu_s$ in uniaxial extension. For many viscoelastic fluids, the extensional viscosity is a more complex function which typically increases with the extension rate [6, 50], a behavior that can also be qualitatively explained with the analogy of a Newtonian solvent laden with polymeric molecules. Under extensional flow, the polymers are stretched, causing additional resistance to further deformation which is "felt" as an increase in viscosity.

There are two observable phenomena that occur in planar and axisymmetric contraction flows that can be partially attributed to extensional viscosity effects. The first is the presence of vortex structures in the salient corners (see Figure 2.5). For a variety of viscoelastic fluids [14] including DNA-laden Newtonian solvents [38], the vortex structure grows rapidly with increasing flow rates, whereas the vortex size remains relatively constant for Newtonian fluids [38]. Second, the entrance length, defined as the distance after the contraction at which the velocity profile reaches 99% of the steady-state Poiseuille profile, has been observed to increase with greater levels of fluid elasticity [39]. The exact mechanisms for these behaviors are not completely understood, however theories have been proposed in which increasing extensional viscosity leads to large pressure drops which affect vortex enhancement [10, 11] and

entrance lengths [39, 73]. Due to the complexity of even simple viscoelastic models, flows in complex geometries such as these contractions are very difficult to solve analytically. It has become apparent that the key to understanding the physics of viscoelastic fluids in contraction geometries, as well as complex geometries in general, lies in numerical simulation.

2.3 The Oldroyd-B Model

The principle of material frame indifference [59] holds that the constitutive laws which govern material behavior should not depend on the external reference frame used to describe them. Oldroyd's [60] foundational work on this subject in the 1950's developed two examples for viscoelastic flow which exhibit material frame indifference. In one, his model "A", the extra-stress is given by covariant components which refer to a reciprocal material basis. In the other, model "B", the extra-stress is given by contravariant components which refer to material basis vectors. Both of Oldroyd's models reduce in the small strain limit to the same linear constitutive theory proposed by experimentalists of that day to describe viscoelasticity. The Oldroyd-A model implies what is now called a lower-convected Maxwell (or covariant-convected) derivative, based on the Cauchy-Green deformation tensor [50, 51], $\mathbf{C} = \mathbf{g}^T \cdot \mathbf{g}$. The Cauchy-Green tensor is a measure of local changes in extension due to deformation, and the lower-convected Maxwell derivative is defined such that $\overset{\vartriangle}{\mathbf{C}} = 0$:

$$\overset{\vartriangle}{\mathbf{t}} = \frac{\partial \mathbf{t}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{t} + (\nabla \mathbf{u})^T \cdot \mathbf{t} + \mathbf{t} \cdot (\nabla \mathbf{u}). \tag{2.10}$$

The Oldroyd-B model implies what is now called an upper-convected Maxwell (or contravariant-convected) derivative, based on the Finger deformation tensor [50, 51], $\mathbf{B} = \mathbf{C}^{-1} = \mathbf{f} \cdot \mathbf{f}^T$. The Finger tensor is a measure of local changes in area due to

deformation, and the upper-convected Maxwell derivative is defined such that $\overset{\nabla}{\mathbf{B}} = 0$:

$$\overset{\nabla}{\mathbf{t}} = \frac{\partial \mathbf{t}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{t} - (\nabla \mathbf{u}) \cdot \mathbf{t} - \mathbf{t} \cdot (\nabla \mathbf{u})^T. \tag{2.11}$$

A review of other frame indifferent models is given in [44].

Recall the relationship between stress and strain in a Maxwell element from (2.3):

$$\frac{\partial \tau_e}{\partial t} = \frac{\mu_e}{\lambda} \frac{\partial \epsilon}{\partial t} - \frac{1}{\lambda} \tau_e. \tag{2.12}$$

Oldroyd replaced the time derivative in the stress-strain relationship for a Maxwell element (2.12) with a convected derivative of the stress tensor and the rate of strain with the rate-of-strain tensor. It has been found that the Oldroyd-B model provides better agreement with experimental observations [34, 63]. In particular, the Oldroyd-B model predicts rod climbing, while the Oldroyd-A model predicts rod depression, that is, the fluid level is depressed at the rod and elevated at the outer wall. Thus, the Oldroyd-B model has become more widely accepted, given by:

$$\overset{\nabla}{\boldsymbol{\tau}} = \frac{\mu_p}{\lambda} 2\mathbf{D} - \frac{1}{\lambda} \boldsymbol{\tau}, \tag{2.13}$$

or equivalently:

$$\frac{\partial \boldsymbol{\tau}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\tau} - (\nabla \mathbf{u}) \cdot \boldsymbol{\tau} - \boldsymbol{\tau} \cdot (\nabla \mathbf{u})^T - \frac{\mu_p}{\lambda}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T] = -\frac{1}{\lambda} \boldsymbol{\tau}. \tag{2.14}$$

The elongational viscosity, μ_e, has been replaced with the polymeric contribution to total viscosity, μ_p, since this particular model is often used to simulate viscoelastic solutions comprising of a viscous solvent with added polymers or biological macromolecules. The tensor $\boldsymbol{\tau}$ shall be referred to as the polymeric stress. The viscosities, μ_s and μ_p, are independent of one another.

Consider the Oldroyd-B model expressed in index notation:

$$\frac{\partial \tau_{ij}}{\partial t} + u_k \frac{\partial \tau_{ij}}{\partial x_k} - \frac{\partial u_i}{\partial x_k}\tau_{kj} - \tau_{ik}\frac{\partial u_j}{\partial x_k} - \frac{\mu_p}{\lambda}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = -\frac{1}{\lambda}\tau_{ij}. \tag{2.15}$$

For steady shear in three dimensions (see Figure 2.2), the polymeric stress profiles can be derived by using equation (2.15) and the conditions:

$$u_1 = u_2 = 0, \qquad \nabla \mathbf{u} = \begin{bmatrix} 0 & \dot{\gamma} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \frac{\partial \boldsymbol{\tau}}{\partial t} = \frac{\partial \boldsymbol{\tau}}{\partial x_0} = \frac{\partial \boldsymbol{\tau}}{\partial x_2} = 0. \tag{2.16}$$

The normal polymeric stresses and the stress differences for the Oldroyd-B model are:

$$\tau_{00} = 2\mu_p \lambda \dot{\gamma}^2, \qquad \tau_{11} = \tau_{22} = 0, \tag{2.17}$$

$$N_1(\dot{\gamma}) = 2\mu_p \lambda \dot{\gamma}^2, \qquad N_2(\dot{\gamma}) = 0. \tag{2.18}$$

Thus, the Oldroyd-B model predicts a quadratic first normal stress difference and zero second normal stress difference. Note that the viscosity remains constant in shear. The lack of shear thinning or thickening is a major downfall of this model when attempting to simulate many real fluids.

For uniaxial extensional flow (see Figure 2.4), the polymeric stress profiles can be derived by using equation (2.15) and the conditions:

$$\mathbf{u} = 0, \qquad \nabla \mathbf{u} = \begin{bmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\frac{\dot{\epsilon}}{2} & 0 \\ 0 & 0 & -\frac{\dot{\epsilon}}{2} \end{bmatrix}, \qquad \frac{\partial \boldsymbol{\tau}}{\partial t} = 0. \tag{2.19}$$

The normal polymeric stresses and the extensional viscosity for the Oldroyd-B model

are:

$$\tau_{00} = \frac{2\mu_p\dot{\epsilon}}{1-2\lambda\dot{\epsilon}}, \qquad \tau_{11} = \tau_{22} = -\frac{\mu_p\dot{\epsilon}}{1+\lambda\dot{\epsilon}}, \tag{2.20}$$

$$\mu_E(\dot{\epsilon}) = \frac{2\mu_p}{1-2\lambda\dot{\epsilon}} + \frac{\mu_p}{1+\lambda\dot{\epsilon}} + 3\mu_s. \tag{2.21}$$

For slow elongation rates ($\lambda\dot{\epsilon} \ll 1$), the extensional viscosity approaches $\mu_E(\dot{\epsilon}) \to 3(\mu_s + \mu_p)$. As $\dot{\epsilon}$ approaches a limiting value in the elongational direction $\dot{\epsilon} \to 1/(2\lambda)$ or compressive direction $\dot{\epsilon} \to -1/\lambda$, the extensional viscosity approaches infinity. Recall that the Oldroyd-B model is based on a spring and dashpot in series. These two cases represent when the spring has become stretched to an infinite length or compressed to zero length. The fact that the Oldroyd-B model permits both an infinite and vanishing spring length leading to an infinite extensional viscosity at finite strain rates is another major downfall of this model. Despite these shortcoming, the Oldroyd-B model is well-suited to model Boger fluids [13], which are very dilute solutions of a high molecular weight polymer in a solvent with high viscosity, since they exhibit elastic effects such as (approximately) quadratic normal stress differences and large extensional viscosities with very little shear thinning or thickening behavior.

The Oldroyd-B model states that the fluid extra-stress consists of a viscous stress due to the solvent, $\boldsymbol{\tau}_s$, and the polymeric stress, $\boldsymbol{\tau}$:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \underbrace{\mu_s\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]}_{\boldsymbol{\tau}_s} + \boldsymbol{\tau}. \tag{2.22}$$

Substituting (2.22) into the differential form of conservation of momentum (1.8) and applying the divergence constraint (1.7) gives rise to the evolutionary form of the incompressible Navier-Stokes equations used in this thesis:

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \frac{1}{\rho}\nabla\cdot\boldsymbol{\tau} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u}. \tag{2.23}$$

The characteristic viscosity used to compute the Reynolds number for this coupled system is now defined as the sum of the solvent and polymeric contributions, $\mu = \mu_s + \mu_p$. A special case of the Oldroyd-B model is the Newtonian limit ($\lambda \to 0$), in which the polymeric stress has the same form as the viscous stress:

$$\boldsymbol{\tau} \to \mu_p 2\mathbf{D} = \mu_p \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \quad \text{as} \quad \lambda \to 0. \tag{2.24}$$

In this case, the system of equations can be expressed as a Newtonian fluid with an augmented viscosity:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{\mu_s + \mu_p}{\rho} \Delta \mathbf{u} - \frac{1}{\rho} \nabla p. \tag{2.25}$$

Equations (2.14) and (2.23) are the evolutionary part of the governing equations, containing both parabolic diffusion terms and hyperbolic advection terms. The diffusion equation is a simple example of a parabolic equation:

$$\frac{\partial \phi}{\partial t} = D \Delta \phi + f. \tag{2.26}$$

The presence of the Laplacian operator in the momentum equation (2.23) accounts for the parabolic nature of the equations. The linear advection equation is a simple example of a hyperbolic equation:

$$\frac{\partial \phi}{\partial t} + a \frac{\partial \phi}{\partial x} = 0. \tag{2.27}$$

The hyperbolic nature of (2.14) and (2.23) can be seen by analyzing the equations expressed in quasi-linear form, which is a multidimensional, multivariable, nonlinear version of (2.27):

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}_0(\mathbf{W}) \cdot \frac{\partial \mathbf{W}}{\partial x_0} + \mathbf{A}_1(\mathbf{W}) \cdot \frac{\partial \mathbf{W}}{\partial x_1} = \mathbf{S}_W(\mathbf{W}), \tag{2.28}$$

where $\mathbf{W} \in \mathbb{R}^{N_W}$ is the vector of primitive variables, $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{R}^{N_W \times N_W}$ are the primitive coefficient matrices, and $\mathbf{S}_W \in \mathbb{R}^{N_W}$ is the primitive source term vector. This system is said to be hyperbolic if $\mathbf{A}_0$ and $\mathbf{A}_1$ contain only real eigenvalues each with a corresponding set of linearly independent eigenvectors. For the model equations, if $\mathbf{W}$ is ordered as follows:

$$\mathbf{W} = \left[u_0, u_1, \tau_{00}, \tau_{10}, \tau_{11}\right]^T, \tag{2.29}$$

then $\mathbf{A}_0$ and $\mathbf{S}_W$ are given as follows:

$$\mathbf{A}_0 = \begin{bmatrix} u_0 & 0 & -\frac{1}{\rho} & 0 & 0 \\ 0 & u_0 & 0 & -\frac{1}{\rho} & 0 \\ -2(\frac{\mu_p}{\lambda} + \tau_{00}) & 0 & u_0 & 0 & 0 \\ -\tau_{10} & -(\frac{\mu_p}{\lambda} + \tau_{00}) & 0 & u_0 & 0 \\ 0 & -2\tau_{10} & 0 & 0 & u_0 \end{bmatrix}, \tag{2.30}$$

$$\mathbf{S}_W = \begin{bmatrix} \mathbf{S}_{W,\mathbf{u}} \\ \mathbf{S}_{W,\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} \\ -\frac{1}{\lambda}\boldsymbol{\tau} \end{bmatrix}. \tag{2.31}$$

The eigenvalues of $\mathbf{A}_0$ are real provided that $\mu_p/\lambda + \tau_{00} > 0$, given as:

$$\boldsymbol{\Lambda}(\mathbf{A}_0) = \left\{ u_0 \pm \sqrt{\frac{2(\frac{\mu_p}{\lambda} + \tau_{00})}{\rho}}, u_0 \pm \sqrt{\frac{\frac{\mu_p}{\lambda} + \tau_{00}}{\rho}}, u_0 \right\}. \tag{2.32}$$

The right column eigenvectors are linearly independent, listed in increasing order as

follows:

$$\mathbf{R}(\mathbf{A}_0) = \begin{bmatrix} -\sqrt{\frac{2(\frac{\mu_p}{\lambda}+\tau_{00})^3}{\rho}} & 0 & 0 & 0 & \sqrt{\frac{2(\frac{\mu_p}{\lambda}+\tau_{00})^3}{\rho}} \\ -\sqrt{\frac{2(\frac{\mu_p}{\lambda}+\tau_{00})}{\rho}}\tau_{10} & -\sqrt{\frac{\mu_p}{\lambda}+\tau_{00}} & 0 & \sqrt{\frac{\mu_p}{\lambda}+\tau_{00}} & \sqrt{\frac{2(\frac{\mu_p}{\lambda}+\tau_{00})}{\rho}}\tau_{10} \\ -2(\frac{\mu_p}{\lambda}+\tau_{00})^2 & 0 & 0 & 0 & -2(\frac{\mu_p}{\lambda}+\tau_{00})^2 \\ -2(\frac{\mu_p}{\lambda}+\tau_{00})\tau_{10} & -(\frac{\mu_p}{\lambda}+\tau_{00}) & 0 & -(\frac{\mu_p}{\lambda}+\tau_{00}) & -2(\frac{\mu_p}{\lambda}+\tau_{00})\tau_{10} \\ -2\tau_{10}^2 & -2\tau_{10} & 1 & -2\tau_{10}^2 & -2\tau_{10} \end{bmatrix}. \tag{2.33}$$

Similarly, the eigenvalues of $\mathbf{A}_1$ can be shown to be real with linearly independent eigenvectors provided that $\mu_p/\lambda + \tau_{11} > 0$, and therefore the system is hyperbolic. Fortunately, it has been proven that the eigenvalues of the system will always be real provided the initial state contains only real eigenvalues [77].

The eigenvalue/eigenvector pairs, referred to as modes, have special names. The eigenvalues $\Lambda = u_0 \pm \sqrt{2(\mu_p/\lambda + \tau_{00})/\rho}$ correspond to longitudinal modes which can be physically described as compressive plane waves [77]. These modes are inherent to the evolutionary equations and cannot physically exist in an incompressible fluid. The approach taken in this thesis is to use a special variable splitting technique to numerically suppress these waves. The eigenvalues $\Lambda = u_0 \pm \sqrt{(\mu_p/\lambda + \tau_{00})/\rho}$ correspond to transverse modes which can be physically described as elastic shear waves. The eigenvalue $\Lambda = u_0$ corresponds to the contact discontinuity, which indicates that certain fluid properties are simply advected with the fluid.

The elastic Mach number is a dimensionless number relating the characteristic fluid velocity to the characteristic speed of elastic waves propagating in the fluid [77], given by:

$$\mathrm{Ma} = \frac{U}{\sqrt{\mu_p/(\rho\lambda)}}. \tag{2.34}$$

The elastic Mach number was originally defined by Ultman and Denn [79] as $\mathrm{Ma} = \sqrt{\mathrm{Re}\cdot\mathrm{We}} = U/\sqrt{\mu/(\rho\lambda)}$. For the Oldroyd-B model, the elastic wave speeds depend

only on μ_p and not μ_s, and therefore definition (2.34) is more relevant to this thesis.

There are more advanced constitutive models for viscoelasticity, many having similar form to the Oldroyd-B model. As mentioned before, the Oldroyd-B model is chosen in this thesis due to its relatively simple structure. Phan-Thien and Tanner [64] replaced the upper-convected Maxwell derivative with a linear combination of the upper and lower-convected Maxwell derivatives, and scaled the decay term:

$$\left(1-\frac{\chi_{\mathrm{PTT}}}{2}\right)\overset{\nabla}{\boldsymbol{\tau}}+\frac{\chi_{\mathrm{PTT}}}{2}\overset{\Delta}{\boldsymbol{\tau}}=\frac{\mu_p}{\lambda}2\mathbf{D}-\frac{1}{\lambda}\boldsymbol{\tau}e^{\epsilon_{\mathrm{PTT}}\lambda/\mu_p};\qquad 0<\chi_{\mathrm{PTT}}<1, \tag{2.35}$$

where χ_{PTT} and ϵ_{PTT} are model parameters. A non-zero choice of χ_{PTT} allows for a non-zero second normal stress difference in steady shear flow. The scaling of the decay term allows for a more realistic extensional viscosity response. Another model is the White-Metzner model [82], which is an Oldroyd-type model with a polymeric viscosity and relaxation time which vary in space and time. The Giesekus model [33] attempts to control the non-physical infinite extensional viscosity in the Oldroyd-B model by incorporating additional nonlinear terms, as in:

$$\overset{\nabla}{\boldsymbol{\tau}}=\frac{\mu_p}{\lambda}2\mathbf{D}-\frac{1}{\lambda}\boldsymbol{\tau}-\frac{\chi_{\mathrm{G}}}{\mu_p}(\boldsymbol{\tau}\cdot\boldsymbol{\tau}), \tag{2.36}$$

where χ_{G} is a model parameter. Since many of these models have similar structure to the Oldroyd-B model, hopefully the techniques discussed in this thesis can be extended to these more realistic models.

3 Numerical Methods

3.1 History of Computational Viscoelasticity

In the early 1970's, computational power had increased to the point where computational scientists could adapt known solution techniques for Newtonian fluids (the incompressible Navier-Stokes equations in particular) to non-Newtonian fluids. Some of the first non-Newtonian simulations were for fluids with variable viscosity designed to predict shear thinning and thickening behavior. In particular, the viscosity was a function of the rate-of-strain tensor. Such problems were easily solved using both finite difference and finite element techniques for Newtonian fluids, with a list of early works provided in [27, 28]. However, the applications of such models were limited, since they did not incorporate any elastic effects found in most non-Newtonian fluids.

Another approach taken by computational scientists at that time was to model viscoelastic behavior using the so-called "hierarchy equations" of Coleman and Noll [26], which were a truncated form of the Rivlin-Ericksen model [69] for viscoelastic stress. In this model, the extra-stress is an explicit function of the velocity and higher-order time derivatives of the velocity. These problems were also easily solved using established finite difference and finite element techniques [27, 28]. However, it was proven early on that the hierarchy equations are only mathematically valid for slow moving, slightly elastic liquids ($\mathrm{We} \ll 1$) [48], a claim that is still widely accepted today [61]. Therefore, this approach was not particularly useful for predicting most real-world viscoelastic phenomena of interest.

The third approach at that time was to model complex flows of highly elastic liquids using Oldroyd-type models. Unfortunately, the application of existing methods for solving Newtonian fluids have not extended easily to Oldroyd-type fluids. By the early 1980's, the so-called "high Weissenberg number problem" had been identified as the outstanding problem in computational rheology [28, 61], in which stable and

convergent numerical solutions of viscoelastic fluids in abrupt geometries (such as planar and axisymmetric contractions) could only be computed up to We ~ 1. The previous example of vortex formation in abrupt contraction geometries is the classic example of a complex flow which suffers from the high We problem. The high We problem has persisted to this day, despite many attempts at using both finite element and finite volume approaches. Even the exact origin of this problem is still an open problem. For a list of early attempts at computing highly elastic complex flows, see [28, 74, 61]. Here, the focus will be on the current state of the problem and modern difficulties in addressing it.

Recently, Aboubacar et al. [1] analyzed the creeping flow (Re ≈ 0) of an Oldroyd-B fluid in a 4:1 planar contraction. By solving the momentum equations with a finite element approach and the stress equations with a finite volume approach, they were able to obtain steady-state convergent results up to We $= 4.4$, an improvement over traditional finite element methods. They attribute the breakdown in the method due to steep stress boundary layers that develop near the contraction, citing the need for further mesh refinement to remedy the problem. Phillips and Williams [65, 66] used a semi-Lagrangian finite volume method using particle tracking for both creeping and inertial (Re ≈ 1) Oldroyd-B fluids in a 4:1 planar contraction. They were able to obtain steady-state convergent results up to We $= 2.5$ and also attribute the loss of convergence to steep gradients that form near the contraction which require further refinement. Alves et al. [4] use a fully implicit finite volume approach to solve creeping flows of Oldroyd-B fluids through planar contractions and obtained convergent results up to De $= 3$. Fattal and Kupferman [31] studied the creeping flow of an Oldroyd-B fluid in a lid driven cavity (an enclosed box with one wall moving with a fixed velocity). They argued that the high We problem could be avoided by using a logarithmic scaling of variables to properly capture exponential stress profiles. They reported convergence for We $= 3$, noting that this was nearly an order of magnitude greater than previous approaches to this problem,

and also believe that increased resolution may lead to improved results.

Joseph et al. [45] take a different approach to the problem and identify the elastic Mach number, Ma, as the critical parameter in understanding viscoelastic flows. They note that for steady flows, Oldroyd-type equations change type from elliptic to hyperbolic in the transition to the supercritical (Ma $>$ 1) regime. They strengthened their argument by citing the experimental results of Ultman and Denn [79], who noticed dramatic changes in flow field and heat transfer properties of dilute polymer solutions when Ma $>$ 1. Joseph makes an analogy to transonic flow, where the characteristic speed of sound becomes greater than the sound speed, and suggested that a time-dependent upwind method for hyperbolic systems could remedy the problem.

Recently, Trebotich, et al. [77] used a Lax-Wendroff finite difference method to simulate time-dependent Oldroyd-B fluids in contraction channels. Steady solutions were not available due to the elastic wave propagation originating at the contraction, yet the unsteady, time-dependent solution was stable and convergent for all We. The only restriction on the algorithm was the requirement of a sub-critical Mach number (Ma $<$ 1). Trebotich hypothesized that previous numerical difficulties were a result of using steady-state or implicit methods, when an appropriately designed unsteady method may be the proper approach. Trebotich also cites the need for a higher-order upwind method to further examine the problem, forming the starting point for the work in this thesis.

3.2 Upwind Methods

Consider the following first-order system of one-dimensional PDEs:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A} \cdot \frac{\partial \mathbf{W}}{\partial x} = 0, \tag{3.1}$$

where the coefficient matrix, $\mathbf{A}$, is constant. As discussed in Section 2.3, a system of this form is said to be hyperbolic if all of the eigenvalues of $\mathbf{A}$ are real and there is a corresponding set of linearly independent eigenvectors. If this is the case, then $\mathbf{W}$ can be uniquely decomposed into a linear combination of the eigenvectors of $\mathbf{A}$. Each of these pieces of $\mathbf{W}$ travels with a velocity equal to its corresponding eigenvalue. The behavior of the solution becomes more complicated for nonlinear equations such as the Oldroyd-B model, where for example, $\mathbf{A}$ is a function of the solution vector, but there is always an underlying wave structure that is essential to the construction of stable numerical algorithms.

Upwind methods are a class of numerical methods for solving hyperbolic PDEs which use only "upstream" information to advance the solution at a given location. The use of "downstream" information to advance a solution is unstable. In 1959, Godunov [36] introduced a conservative finite difference upwind method for one-dimensional ideal gas dynamics, which is a system of nonlinear hyperbolic PDEs. This work was first-order accurate and has become the foundation for many modern upwind-based numerical methods. At that time, traditional finite difference methods operated under the assumption that the solution was smoothly varying everywhere and relied primarily on Taylor series analysis. In contrast, Godunov represented solutions as piecewise constant states, and the interaction of these states gave rise to interface values that were used to advance the solution. A key feature in all Godunov methods is the use of a Riemann solver, which computes the future behavior of a system consisting of two states initially separated by a jump discontinuity. Godunov [37] also presented a two-dimensional version of his method in an Eulerian framework, rather than the Lagrangian framework in the original paper. Godunov-type methods have proven to be more robust than traditional finite difference methods in the presence of discontinuities that can occur in hyperbolic systems [83].

Van Leer [80] extended Godunov's method to second-order in Lagrangian coordinates by representing the solution as a collection of piecewise linear functions. A

slope limiting procedure was used to prevent the buildup of oscillations at discontinuities which were otherwise unavoidable in explicit second-order methods [36]. An operator splitting technique was used which involved an alternating series of one-dimensional solutions to advance the full-dimensional solution. Colella presented a version of van Leer's work in an Eulerian framework using both piecewise linear [20, 21] and piecewise parabolic [25] representations of the solution. Later, Colella presented an unsplit method [22], which was later extended to an embedded boundary (EB) geometry representation [24]. The model equations have traditionally been the Euler equations for ideal gas dynamics, chosen for their relatively simple, yet nonlinear behavior. In more recent works [22, 24], a methodology for general systems of hyperbolic conservation laws has been proposed. The overall time-stepping strategy in this thesis is based on the multi-dimensional, EB version of Godunov's method in [24].

There are two immediate advantages for using this Godunov method over the Lax-Wendroff discretization used by Trebotich [77]. The first advantage is the ability to handle arbitrary smooth irregular geometries. Second, the Godunov method increases the allowable time step by a factor of four, as it is subject to following advective Courant-Friedrichs-Lewy (CFL) time step constraint required for stability [49]:

$$\Delta t \leq \mathrm{CFL}\frac{h}{|\Lambda_{\max}|}; \qquad 0 < \mathrm{CFL} < 1, \tag{3.2}$$

where h is the grid spacing,"CFL" is the CFL number, and $\Lambda_{\max}$ is the eigenvalue of largest magnitude in the primitive coefficient matrices over the entire domain, including the boundary conditions.

Unfortunately, the time-stepping strategy in [24] cannot be applied directly to the equations in this thesis. First, the equations are not purely hyperbolic, as the primitive variable form contains non-zero source terms. Second, the fluid is incompressible, requiring additional steps for its enforcement. Also, as is the case

with any problem, the appropriate boundary conditions and Riemann solver must be derived.

3.3 Projection Methods

Chorin [17] introduced a numerical method for the incompressible Navier-Stokes equations. Chorin's projection method is based on the Hodge decomposition of a vector field [18], in which an arbitrary vector field, in this case the velocity, on a domain, Ω, can be uniquely decomposed into a part that is divergence-free and a part that is curl-free (represented as the gradient of a scalar field):

$$\mathbf{v} = \mathbf{v}_{\mathrm{d}} + \nabla\phi, \tag{3.3}$$

with boundary conditions $\mathbf{v}_{\mathrm{d}} \cdot \mathbf{n} = 0$ and $\mathbf{n} \cdot \nabla\phi = \mathbf{v} \cdot \mathbf{n}$ on $\partial\Omega$. By taking the divergence of each side of this equation, an elliptic Poisson's equation is revealed:

$$\nabla \cdot \mathbf{v} = \nabla \cdot \mathbf{v}_{\mathrm{d}} + \nabla \cdot \nabla\phi, \tag{3.4}$$

$$\nabla \cdot \mathbf{v} = \Delta\phi. \tag{3.5}$$

Once ϕ has been computed, the divergence-free velocity is found by using:

$$\mathbf{v}_{\mathrm{d}} = \mathbf{v} - \nabla\phi. \tag{3.6}$$

In Chorin's original method [17], the velocity field is advanced in time using the evolutionary part of the incompressible Navier-Stokes equations. This preliminary velocity is projected to enforce the divergence constraint and the curl-free term is used to update the pressure. Bell, Colella, and Glaz [8, 58] extended Chorin's method to second-order accuracy in space and time by utilizing a second-order Godunov method to capture the hyperbolic advection terms. Almgren, et al. [3]

introduced an approximate projection method for node-centered pressures with a compact Laplacian stencil which prevented grid decoupling and simplified the linear algebra required to solve the Poisson equation. The resulting method did not return a discretely divergence-free velocity. Lai [46] introduced an approximate projection method for cell-centered pressures which used an additional filtering step to eliminate non-physical oscillatory modes that persisted due to the approximate nature of the projection. In this thesis, the pressure is cell-centered, and the cell-centered velocity is projected using the ideas in [46].

4 Numerical Details

4.1 Overview of the Numerical Method

The solution domain is divided into rectangular cells, denoted $C_{ij} = C_{\mathbf{i}}$, where the subscript is the spatial location index, and the grid spacing is uniform in all directions such that $h = \Delta x_0 = \Delta x_1$. The state variables are defined at the geometric centers of each cell, and represent the average value over the cell, as illustrated in In Figure 4.1. To be consistent with previous methods involving this geometry description, cells are said to have "volume", and cell edges are referred to as faces which have "area", despite the fact that this is a two-dimensional problem.

In order to advance the solution in time, the equations must first be expressed in a hybrid conservation form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}_U(\mathbf{U}); \qquad \nabla \cdot \mathbf{F}(\mathbf{U}) = \frac{\partial \mathbf{F}_0(\mathbf{U})}{\partial x_0} + \frac{\partial \mathbf{F}_1(\mathbf{U})}{\partial x_1}, \tag{4.1}$$

where $\mathbf{U} \in \mathbb{R}^{N_U}$ is the conserved variable vector, $\mathbf{S}_U \in \mathbb{R}^{N_U}$ is the conserved source term vector, $\mathbf{F} \in \mathbb{R}^{2 \times N_U}$ is the tensor of fluxes, and the rows of $\mathbf{F}$ are the directional flux vectors, $\mathbf{F}_0, \mathbf{F}_1 \in \mathbb{R}^{N_U}$. The basic idea is to discretize the time derivative in

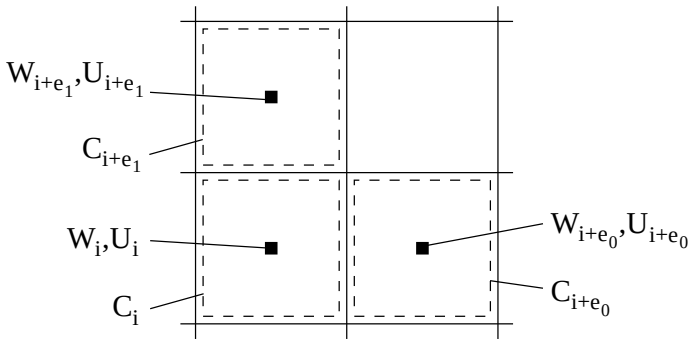

Figure 4.1: Cells and their associated state variables.

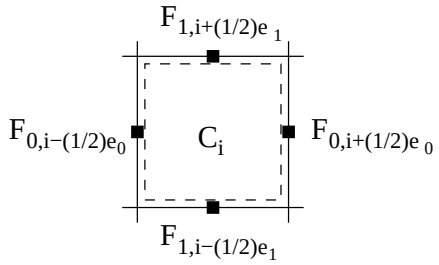

Figure 4.2: Stencil points for the four-point divergence.

(4.1) to form an update equation for $\mathbf{U}$ in each cell:

$$\frac{\partial \mathbf{U}}{\partial t} = \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} \quad \rightarrow \quad \mathbf{U}_{\mathbf{i}}^{n+1} = \mathbf{U}_{\mathbf{i}}^n + \Delta t \left[-(\nabla \cdot \mathbf{F})_{\mathbf{i}} + \mathbf{S}_{U,\mathbf{i}} \right], \tag{4.2}$$

where the superscript is the time step index. For each cell, the divergence can be represented as an integral sum of normal fluxes at the cell faces, $\partial C_{\mathbf{i}}$, using the divergence theorem:

$$(\nabla \cdot \mathbf{F})_{\mathbf{i}} = \frac{1}{|C_{\mathbf{i}}|} \int_{C_{\mathbf{i}}} (\nabla \cdot \mathbf{F}) dV = \frac{1}{|C_{\mathbf{i}}|} \int_{\partial C_{\mathbf{i}}} (\mathbf{n} \cdot \mathbf{F}) dA, \tag{4.3}$$

where $|C_{\mathbf{i}}|$ is the volume of cell $C_{\mathbf{i}}$, and $\mathbf{n}$ is the outward normal of the cell face. Therefore, a discrete approximation of the divergence is the following four-point difference approximation, illustrated in Figure 4.2:

$$(\nabla \cdot \mathbf{F})_{\mathbf{i}} = \frac{1}{h} \sum_{d=0}^{1} \sum_{\pm=+}^{-} \pm \mathbf{F}_{d,\mathbf{i} \pm \mathbf{e}_d} = \frac{\mathbf{F}_{0,\mathbf{i}+\frac{1}{2}\mathbf{e}_0} - \mathbf{F}_{0,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}}{h} + \frac{\mathbf{F}_{1,\mathbf{i}+\frac{1}{2}\mathbf{e}_1} - \mathbf{F}_{1,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}}{h}, \tag{4.4}$$

where $\mathbf{e}_d$ is the d^{th} unit vector.

In order to form a second-order method, the fluxes must be time-centered, that is, they must be computed using the conserved variables at time $t^{n+\frac{1}{2}} = t^n + \Delta t/2$. The face-centered, time-centered estimates of the state variables, referred to as the "marker and cell" (MAC) [41] MAC state, are computed using a reconstruction algorithm followed by a MAC projection. The reconstruction step uses an upwind tech-

nique involving the primitive variable form (2.28) along with Taylor series extrapolations in space and time to obtain a preliminary MAC state. A MAC projection is applied to the preliminary MAC velocity, which enforces the divergence constraint by extracting the part of the velocity field containing numerical divergence, yielding the final MAC state. After obtaining the final MAC state, the cells are updated conservatively using the intercell fluxes. Advanced discretization stencils are used to account for the conservative source terms which result in an overall second-order method which captures the Newtonian and elastic limits of the Oldroyd-B fluid. A cell-centered projection is performed on the preliminary cell-centered velocity field to enforce the divergence constraint.

An EB approach will be used to represent irregular geometry in which rectangular cells are cut into irregular control volumes. For each part of the algorithm, the numerical method will be described under the assumption that the domain is regular (all-fluid), that is, there are no EBs. Next, the modifications for cells near EBs and/or domain boundaries will be described. A domain boundary can have either an inflow, outflow, or solid wall boundary condition, whereas EBs always represents a solid wall. For problems with inflow, the flow is directed in the $+x_0$ direction.

4.1.1 Numerical Stress Splitting

Recall the eigenvalues of the primitive variable formulation from (2.32):

$$\mathbf{\Lambda}(\mathbf{A}_0) = \left\{ u_0 \pm \sqrt{\frac{2(\frac{\mu_p}{\lambda} + \tau_{00})}{\rho}}, u_0 \pm \sqrt{\frac{\frac{\mu_p}{\lambda} + \tau_{00}}{\rho}}, u_0 \right\}. \tag{4.5}$$

These eigenvalues pose an immediate problem. Since an upwind method is being used to compute the intercell fluxes, the CFL time step constraint from (3.2) applies here:

$$\Delta t \leq \mathrm{CFL}\frac{h}{|\Lambda_{\max}|}; \qquad 0 < \mathrm{CFL} < 1. \tag{4.6}$$

In the Newtonian limit ($\lambda \to 0$), the eigenvalues become very large, forcing the time step to become vanishingly small. Using the stress splitting technique in [77], a wave speed a^2 is introduced, in which the Oldroyd-B equations are expressed as:

$$\frac{\partial \boldsymbol{\tau}}{\partial t} + (\mathbf{u}\cdot\nabla)\boldsymbol{\tau} - (\nabla\mathbf{u})\cdot\boldsymbol{\tau} - \boldsymbol{\tau}\cdot(\nabla\mathbf{u})^T - a^2 2\mathbf{D} = \left(\frac{\mu_p}{\lambda} - a^2\right) 2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau}. \tag{4.7}$$

At the beginning of each time step, the wave speed a^2 is initialized to a value between μ_p/λ in the elastic limit ($\lambda \to \infty$, μ_p/λ finite) and 2τ in the Newtonian limit ($\lambda \to 0$), where τ is defined as the absolute value of the global minimum of the normal stresses, $\tau = |\min_d(\tau_{dd})|$. In particular:

$$a^2 = \min\left[\chi(\lambda)a_\infty^2 + [1-\chi(\lambda)]\, a_0^2,\ \frac{\mu_p}{\lambda}\right]; \tag{4.8}$$

$$a_\infty^2 = \lim_{\lambda\to\infty} a^2 = \frac{\mu_p}{\lambda}, \qquad a_0^2 = \lim_{\lambda\to 0} a^2 = 2\tau, \tag{4.9}$$

$$\chi(\lambda) = \frac{\lambda}{t_{\text{adv}}}\left(1 - e^{-\lambda/(2t_{\text{adv}})}\right)\left(1 - e^{-t_{\text{adv}}/\lambda}\right); \tag{4.10}$$

$$t_{\text{adv}} = \frac{h}{|\mathbf{u}|_\infty}, \tag{4.11}$$

where $|\mathbf{u}|_\infty$ is the maximum fluid speed in the domain. Essentially, a^2 gives some control over which terms will be discretized with the upwind method. The matrix $\mathbf{A}_0$ and vector $\mathbf{S}_W$ are now:

$$\mathbf{A}_0 = \begin{bmatrix} u_0 & 0 & -\frac{1}{\rho} & 0 & 0 \\ 0 & u_0 & 0 & -\frac{1}{\rho} & 0 \\ -2\alpha_0 & 0 & u_0 & 0 & 0 \\ -\tau_{10} & -\alpha_0 & 0 & u_0 & 0 \\ 0 & -2\tau_{10} & 0 & 0 & u_0 \end{bmatrix}, \tag{4.12}$$

$$\mathbf{S}_W = \begin{bmatrix} \mathbf{S}_{W,\mathbf{u}} \\ \mathbf{S}_{W,\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} \\ \left(\frac{\mu_p}{\lambda} - a^2\right) 2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau} \end{bmatrix}, \tag{4.13}$$

where $\alpha_d = a^2 + \tau_{dd}$. The eigenvalues of (4.12) in increasing order are:

$$\boldsymbol{\Lambda}(\mathbf{A}_0) = \left\{ u_0 - \sqrt{2}c_0, u_0 - c_0, u_0, u_0 + c_0, u_0 + \sqrt{2}c_0 \right\}, \tag{4.14}$$

where $c_d = \sqrt{(a^2 + \tau_{dd})/\rho}$. In [77], it is shown that this stress splitting can increase the time step in the Newtonian limit by a factor of 10^6 or greater.

4.1.2 Conservation Forms

The equations of motion are now cast into conservation form (4.1). For the incompressible Navier-Stokes equations (1.13), use the following identity:

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot(\mathbf{u}\mathbf{u}) - \mathbf{u}\underbrace{(\nabla\cdot\mathbf{u})}_{0}. \tag{4.15}$$

The conservation form follows directly:

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\mathbf{u} - \frac{1}{\rho}\boldsymbol{\tau}) = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u}. \tag{4.16}$$

In order to derive conservation equations involving the polymeric stress tensor, the inverse deformation gradient, $\mathbf{g}$, needs to be coupled into our system of equations. Begin with the identity from (1.21):

$$\nabla\mathbf{u} = -\mathbf{f}\cdot\frac{D\mathbf{g}}{Dt}. \tag{4.17}$$

Substitute this equation into the left hand side of the Oldroyd-B equations (4.7) to

get:

$$\frac{D\boldsymbol{\tau}}{Dt} + \underbrace{\mathbf{f}\cdot\frac{D\mathbf{g}}{Dt}}_{-\nabla\mathbf{u}}\cdot\boldsymbol{\tau} + \boldsymbol{\tau}\cdot\underbrace{\frac{D\mathbf{g}^T}{Dt}\cdot\mathbf{f}^T}_{-\nabla\mathbf{u}^T} + a^2\left(\underbrace{\mathbf{f}\cdot\frac{D\mathbf{g}}{Dt}}_{-\nabla\mathbf{u}} + \underbrace{\frac{D\mathbf{g}^T}{Dt}\cdot\mathbf{f}^T}_{-\nabla\mathbf{u}^T}\right) = \left(\frac{\mu_p}{\lambda} - a^2\right)2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau}. \tag{4.18}$$

Pre-multiply each side of this equation by $\mathbf{g}$ and post-multiply by $\mathbf{g}^T$ to get:

$$\begin{aligned}\mathbf{g}\cdot\frac{D\boldsymbol{\tau}}{Dt}\cdot\mathbf{g}^T + \frac{D\mathbf{g}}{Dt}\cdot\boldsymbol{\tau}\cdot\mathbf{g}^T + \mathbf{g}\cdot\boldsymbol{\tau}\cdot\frac{D\mathbf{g}^T}{Dt} + a^2\left(\frac{D\mathbf{g}}{Dt}\cdot\mathbf{g}^T + \mathbf{g}\cdot\frac{D\mathbf{g}^T}{Dt}\right)& \\ = \mathbf{g}\cdot\left[\left(\frac{\mu_p}{\lambda} - a^2\right)2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau}\right]\cdot\mathbf{g}^T.&\end{aligned} \tag{4.19}$$

The terms on the left hand side (LHS) of this equation can be collected and simplified as follows:

$$\begin{aligned}\text{LHS} &= \left[\mathbf{g}\cdot\frac{D\boldsymbol{\tau}}{Dt}\cdot\mathbf{g}^T + \frac{D\mathbf{g}}{Dt}\cdot\boldsymbol{\tau}\cdot\mathbf{g}^T + \mathbf{g}\cdot\boldsymbol{\tau}\cdot\frac{D\mathbf{g}^T}{Dt}\right] + \left[a^2\left(\frac{D\mathbf{g}}{Dt}\cdot\mathbf{g}^T + \mathbf{g}\cdot\frac{D\mathbf{g}^T}{Dt}\right)\right] \\ &= \frac{D}{Dt}\left(\mathbf{g}\cdot\boldsymbol{\tau}\cdot\mathbf{g}^T\right) + \frac{D}{Dt}\left(\mathbf{g}\cdot a^2\mathbf{I}\cdot\mathbf{g}^T\right) \\ &= \frac{D}{Dt}\left(\mathbf{g}\cdot(\boldsymbol{\tau} + a^2\mathbf{I})\cdot\mathbf{g}^T\right) \\ &= \frac{\partial}{\partial t}\left(\mathbf{g}\cdot(\boldsymbol{\tau} + a^2\mathbf{I})\cdot\mathbf{g}^T\right) + (\mathbf{u}\cdot\nabla)\left(\mathbf{g}\cdot(\boldsymbol{\tau} + a^2\mathbf{I})\cdot\mathbf{g}^T\right).\end{aligned} \tag{4.20}$$

Define a new conserved variable, $\mathbf{M} = \mathbf{g}\cdot(\boldsymbol{\tau} + a^2\mathbf{I})\cdot\mathbf{g}^T$, and use a variant of identity (4.15):

$$(\mathbf{u}\cdot\nabla)\mathbf{M} = \nabla\cdot(\mathbf{u}\mathbf{M}) - \mathbf{M}\underbrace{(\nabla\cdot\mathbf{u})}_{0}. \tag{4.21}$$

The conservation form follows directly:

$$\frac{\partial\mathbf{M}}{\partial t} + \nabla\cdot(\mathbf{u}\mathbf{M}) = \mathbf{g}\cdot\left[\left(\frac{\mu_p}{\lambda} - a^2\right)2\mathbf{D} - \frac{1}{\lambda}\boldsymbol{\tau}\right]\cdot\mathbf{g}^T. \tag{4.22}$$

The inverse deformation gradient is added to both the the vector of primitive and conserved variables. The evolution equations for $\mathbf{g}$ were previously given in (1.22).

To obtain a conservation form for $\mathbf{g}$, first consider the expanded form of the evolution equations:

$$\frac{\partial g_{00}}{\partial t} + \left(u_0 \frac{\partial g_{00}}{\partial x_0} + g_{00} \frac{\partial u_0}{\partial x_0} + g_{01} \frac{\partial u_1}{\partial x_0} \right) + \left(u_1 \frac{\partial g_{00}}{\partial x_1} \right) = 0, \tag{4.23}$$

$$\frac{\partial g_{10}}{\partial t} + \left(u_0 \frac{\partial g_{10}}{\partial x_0} + g_{10} \frac{\partial u_0}{\partial x_0} + g_{11} \frac{\partial u_1}{\partial x_0} \right) + \left(u_1 \frac{\partial g_{10}}{\partial x_1} \right) = 0, \tag{4.24}$$

$$\frac{\partial g_{01}}{\partial t} + \left(u_0 \frac{\partial g_{01}}{\partial x_0} \right) + \left(u_1 \frac{\partial g_{01}}{\partial x_1} + g_{00} \frac{\partial u_0}{\partial x_1} + g_{01} \frac{\partial u_1}{\partial x_1} \right) = 0, \tag{4.25}$$

$$\frac{\partial g_{11}}{\partial t} + \left(u_0 \frac{\partial g_{11}}{\partial x_0} \right) + \left(u_1 \frac{\partial g_{11}}{\partial x_1} + g_{10} \frac{\partial u_0}{\partial x_1} + g_{11} \frac{\partial u_1}{\partial x_1} \right) = 0. \tag{4.26}$$

Rearrange these equations as follows:

$$\frac{\partial g_{00}}{\partial t} + \left(u_0 \frac{\partial g_{00}}{\partial x_0} + g_{00} \frac{\partial u_0}{\partial x_0} + u_1 \frac{\partial g_{01}}{\partial x_0} + g_{01} \frac{\partial u_1}{\partial x_0} \right) = u_1 \left(\frac{\partial g_{01}}{\partial x_0} - \frac{\partial g_{00}}{\partial x_1} \right), \tag{4.27}$$

$$\frac{\partial g_{10}}{\partial t} + \left(u_0 \frac{\partial g_{10}}{\partial x_0} + g_{10} \frac{\partial u_0}{\partial x_0} + u_1 \frac{\partial g_{11}}{\partial x_0} + g_{11} \frac{\partial u_1}{\partial x_0} \right) = u_1 \left(\frac{\partial g_{11}}{\partial x_0} - \frac{\partial g_{10}}{\partial x_1} \right), \tag{4.28}$$

$$\frac{\partial g_{01}}{\partial t} + \left(u_0 \frac{\partial g_{00}}{\partial x_1} + g_{00} \frac{\partial u_0}{\partial x_1} + u_1 \frac{\partial g_{01}}{\partial x_1} + g_{01} \frac{\partial u_1}{\partial x_1} \right) = u_0 \left(\frac{\partial g_{00}}{\partial x_1} - \frac{\partial g_{01}}{\partial x_0} \right), \tag{4.29}$$

$$\frac{\partial g_{11}}{\partial t} + \left(u_0 \frac{\partial g_{10}}{\partial x_1} + g_{10} \frac{\partial u_0}{\partial x_1} + u_1 \frac{\partial g_{11}}{\partial x_1} + g_{11} \frac{\partial u_1}{\partial x_1} \right) = u_0 \left(\frac{\partial g_{10}}{\partial x_1} - \frac{\partial g_{11}}{\partial x_0} \right). \tag{4.30}$$

The right hand side of equations (4.27)-(4.30) is a two-dimensional version of the 3×3 tensor $\left[\mathbf{u} \times (\nabla \times \mathbf{g}^T) \right]^T$ ignoring any terms with a u_2 or x_2 component. Therefore the conservation form is:

$$\frac{\partial}{\partial t}(\mathbf{g} \cdot \mathbf{e}_d) + \nabla \cdot (\mathbf{e}_d \mathbf{g} \cdot \mathbf{u}) = \left[\mathbf{u} \times (\nabla \times \mathbf{g}^T) \right]^T \cdot \mathbf{e}_d; \qquad d = 0, 1, \tag{4.31}$$

The expanded conservation forms are:

$$\frac{\partial}{\partial t}\begin{bmatrix}\mathbf{u}\\ \mathbf{M}\\ \mathbf{g}\cdot\mathbf{e}_0\\ \mathbf{g}\cdot\mathbf{e}_1\end{bmatrix}+\frac{\partial}{\partial x_0}\begin{bmatrix}u_0\mathbf{u}-\frac{1}{\rho}\boldsymbol{\tau}\cdot\mathbf{e}_0\\ u_0\mathbf{M}\\ \mathbf{g}\cdot\mathbf{u}\\ 0\end{bmatrix}+\frac{\partial}{\partial x_1}\begin{bmatrix}u_1\mathbf{u}-\frac{1}{\rho}\boldsymbol{\tau}\cdot\mathbf{e}_1\\ u_1\mathbf{M}\\ 0\\ \mathbf{g}\cdot\mathbf{u}\end{bmatrix}=\mathbf{S}_U, \quad (4.32)$$

$$\mathbf{S}_U=\begin{bmatrix}\mathbf{S}_{U,\mathbf{u}}\\ \mathbf{S}_{U,\mathbf{M}}\\ \mathbf{S}_{U,\mathbf{g}}\end{bmatrix}=\begin{bmatrix}-\frac{1}{\rho}\nabla p+\nu\Delta\mathbf{u}\\ \mathbf{g}\cdot\left[\left(\frac{\mu_p}{\lambda}-a^2\right)2\mathbf{D}-\frac{1}{\lambda}\boldsymbol{\tau}\right]\cdot\mathbf{g}^T\\ \left[\mathbf{u}\times\left(\nabla\times\mathbf{g}^T\right)\right]^T\end{bmatrix}. \quad (4.33)$$

The primitive variable vector is now given as:

$$\mathbf{W}=\left[u_0,u_1,\tau_{00},\tau_{10},\tau_{11},g_{00},g_{10},g_{01},g_{11}\right]^T, \quad (4.34)$$

with $\mathbf{A}_0$ and $\mathbf{S}_W$ given as:

$$\mathbf{A}_0=\begin{bmatrix}u_0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & u_0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0\\ -2\alpha_0 & 0 & u_0 & 0 & 0 & 0 & 0 & 0 & 0\\ -\tau_{10} & -\alpha_0 & 0 & u_0 & 0 & 0 & 0 & 0 & 0\\ 0 & -2\tau_{10} & 0 & 0 & u_0 & 0 & 0 & 0 & 0\\ g_{00} & g_{01} & 0 & 0 & 0 & u_0 & 0 & 0 & 0\\ g_{10} & g_{11} & 0 & 0 & 0 & 0 & u_0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_0\end{bmatrix}, \quad (4.35)$$

$$\mathbf{S}_W=\begin{bmatrix}\mathbf{S}_{W,\mathbf{u}}\\ \mathbf{S}_{W,\boldsymbol{\tau}}\\ \mathbf{S}_{W,\mathbf{g}}\end{bmatrix}=\begin{bmatrix}-\frac{1}{\rho}\nabla p+\nu\Delta\mathbf{u}\\ \left(\frac{\mu_p}{\lambda}-a^2\right)2\mathbf{D}-\frac{1}{\lambda}\boldsymbol{\tau}\\ 0\end{bmatrix}. \quad (4.36)$$

The system is still hyperbolic, as can be seen by the eigenvalues and the matrix of right column eigenvectors:

$$\mathbf{\Lambda}(\mathbf{A}_0) = \left\{ u_0 - \sqrt{2}c_0, u_0 - c_0, \underbrace{u_0, u_0, u_0, u_0, u_0}_{\text{5-fold degenerate}}, u_0 + c_0, u_0 + \sqrt{2}c_0 \right\}, \tag{4.37}$$

$$\mathbf{R}_0 = \begin{bmatrix} -\sqrt{2}\alpha_0 c_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2}\alpha_0 c_0 \\ -\sqrt{2}c_0\tau_{10} & -c_0 & 0 & 0 & 0 & 0 & 0 & c_0 & \sqrt{2}c_0\tau_{10} \\ -2\alpha_0^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\alpha_0^2 \\ -2\alpha_0\tau_{10} & -\alpha_0 & 0 & 0 & 0 & 0 & 0 & -\alpha_0 & -2\alpha_0\tau_{10} \\ -2\tau_{10}^2 & -2\tau_{10} & 1 & 0 & 0 & 0 & 0 & -2\tau_{10} & -2\tau_{10}^2 \\ g_{00}\alpha_0 + g_{01}\tau_{10} & g_{01} & 0 & 1 & 0 & 0 & 0 & g_{01} & g_{00}\alpha_0 + g_{01}\tau_{10} \\ g_{10}\alpha_0 + g_{11}\tau_{10} & g_{11} & 0 & 0 & 1 & 0 & 0 & g_{11} & g_{10}\alpha_0 + g_{11}\tau_{10} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.38}$$

4.2 Embedded Boundary Geometry Representation

The underlying description of space is a set of rectangular Cartesian cells:

$$C_{\mathbf{i}} = [(\mathbf{i} - \frac{1}{2}\boldsymbol{\upsilon})h, (\mathbf{i} + \frac{1}{2}\boldsymbol{\upsilon})h]; \qquad \mathbf{i} \in [0, \mathbf{i}_{\max}], \tag{4.39}$$

where $\boldsymbol{\upsilon}$ is the vector whose entries are all one. Given an irregular solution domain, Ω, the control volumes are denoted $V_{\mathbf{i}} = \Omega \cap C_{\mathbf{i}}$, as illustrated in Figure 4.3. For regular cells, $V_{\mathbf{i}} = C_{\mathbf{i}}$. Control volume faces are the intersection of the control volume edges, $\partial V_{\mathbf{i}}$, with $\partial C_{\mathbf{i}}$ and are denoted $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, as illustrated in Figure 4.4. The EB faces are the intersection of the domain boundary, $\partial\Omega$, with the Cartesian cells, denoted $A_{\mathbf{i},EB} = \partial\Omega \cap C_{\mathbf{i}}$. For this thesis, it is assumed that the irregular domain representation requires at most one control volume and at most one EB face per Cartesian cell, even though the method can be generalized to multiple

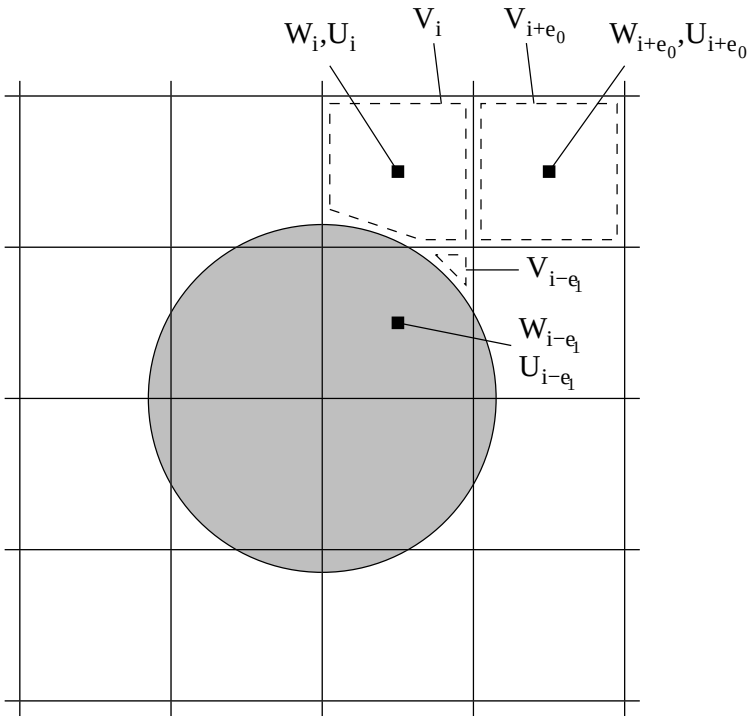

Figure 4.3: Irregular control volumes are created when an irregular domain, in this case a channel containing a solid sphere, is represented on a set of Cartesian cells. The state variables are defined at the center of each Cartesian cell, even if the cell center is outside of the solution domain.

valid control volumes per Cartesian cell provided there is at most one EB face per control volume. To construct conservative finite difference methods using this description, several quantities are needed, all of which can be computed to $\mathcal{O}(h^2)$ in two dimensions by representing the EB as a line segment connecting the two points where $\partial\Omega$ intersects $\partial C_{\mathbf{i}}$:

- The volume fraction of the control volume, $\kappa_{\mathbf{i}} \in [0,1]$, the area fractions of each face, $\alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \in [0,1]$, and the area fraction of the EB face, $\alpha_{\mathbf{i},EB} \in [0,\sqrt{2}]$:

$$\kappa_{\mathbf{i}} = \frac{|V_{\mathbf{i}}|}{h^2}, \qquad \alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \frac{\left|A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}\right|}{h}, \qquad \alpha_{\mathbf{i},EB} = \frac{|A_{\mathbf{i},EB}|}{h}. \tag{4.40}$$

- The location of each face-centroid, $\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, and the normalized location of each

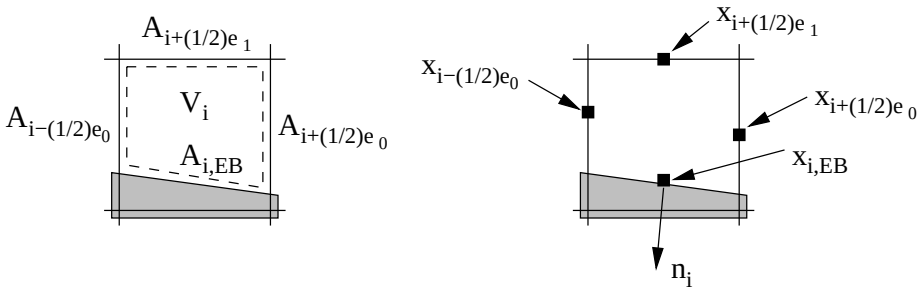

Figure 4.4: (Left) Control volume $V_{\mathbf{i}}$ with faces $A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ and EB face $A_{\mathbf{i},EB}$. (Right) Face-centroids $\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, EB centroid $\mathbf{x}_{\mathbf{i},EB}$, and outward normal $\mathbf{n}_{\mathbf{i}}$.

face-centroid with respect its corresponding Cartesian face-center, $\hat{\mathbf{x}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \in [-\frac{1}{2},\frac{1}{2}]$:

$$\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \frac{1}{\left|A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}\right|}\int_{A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}} \mathbf{x}dA, \qquad \hat{\mathbf{x}}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \frac{1}{h}\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} - \left(\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d\right). \quad (4.41)$$

- The location of the EB centroid, $\mathbf{x}_{\mathbf{i},EB}$, and the normalized location of the EB centroid with respect to its corresponding Cartesian cell-center, $\hat{\mathbf{x}}_{\mathbf{i},EB} \in [-\frac{1}{2}\boldsymbol{\upsilon},\frac{1}{2}\boldsymbol{\upsilon}]$:

$$\mathbf{x}_{\mathbf{i},EB} = \frac{1}{|A_{\mathbf{i},EB}|}\int_{A_{\mathbf{i},EB}} \mathbf{x}dA, \qquad \hat{\mathbf{x}}_{\mathbf{i},EB} = \frac{1}{h}\mathbf{x}_{\mathbf{i},EB} - \mathbf{i}. \quad (4.42)$$

- The normalized average outward normal of $\partial\Omega$ over $A_{\mathbf{i},EB}$, $\mathbf{n}_{\mathbf{i}}$:

$$\mathbf{n}_{\mathbf{i}} = \frac{1}{|A_{\mathbf{i},EB}|}\int_{A_{\mathbf{i},EB}} \mathbf{n}dA, \qquad |\mathbf{n}_{\mathbf{i}}| = 1. \quad (4.43)$$

Each Cartesian cell is classified as open or covered. Covered cells lie completely within the EB and are not part of the solution domain. Open cells are classified as either regular or irregular. Regular cells are completely uncovered and irregular cells are partially covered. A cell is considered to be irregular even if the EB only shares a portion of the cell boundary and does not actually come inside the cell.

Faces are classified in the same way. For notational simplicity in this thesis, it is assumed that if a face is covered, then at least one of the cells adjacent to the face is covered. This condition is equivalent to assuming a minimum thickness to the covered regions.

State variables are defined at the geometric centers of each open cell, even if the cell center is outside the solution domain. The cell-centered values in irregular cells represent smooth extensions of the state variables to the cell-center, which is permitted as long as the solution and the domain boundary are sufficiently smooth [35]. There are several advantages to using a cell-centered representation rather than a centroid-centered representation. The regular structure of the data leads to better control over truncation error, and the resulting finite difference operators are more amenable to multigrid and adaptive methods [23].

Using the geometric quantities defined above, the four-point divergence for regular cells (4.3) can be generalized to any control volume. The "conservative" divergence of a vector field [24] associated with control volume $V_{\mathbf{i}}$ can be expressed as a linear combination of face-centered and EB centroid normal components using the divergence theorem:

$$\begin{aligned}(\nabla \cdot \mathbf{v})_{\mathbf{i}}^{C,FC\rightarrow CC} &= \frac{1}{|V_{\mathbf{i}}|}\int_{V_{\mathbf{i}}}(\nabla\cdot\mathbf{v})dV = \frac{1}{|V_{\mathbf{i}}|}\int_{\partial V_{\mathbf{i}}}(\mathbf{n}\cdot\mathbf{v})dA \\ &\approx \frac{1}{\kappa_{\mathbf{i}}h}\left[\sum_{d=0}^{1}\sum_{\pm=+,-}\pm\alpha_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}v_d(\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}) + \alpha_{\mathbf{i},EB}\mathbf{n}_{\mathbf{i}}\cdot\mathbf{v}_{\mathbf{i},EB}\right].\end{aligned} \tag{4.44}$$

where $v_d(\mathbf{x}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d})$ is the value of v_d at face-centroids, obtained by linearly interpolating between the two nearest face-centered values of v_d, as illustrated in Figure 4.5. Figure 4.6 illustrates a sample conservative divergence discretization.

In contrast to (4.44) is the "non-conservative" divergence [24], which is the following linear combination of face-centered values of v_d, regardless of whether the

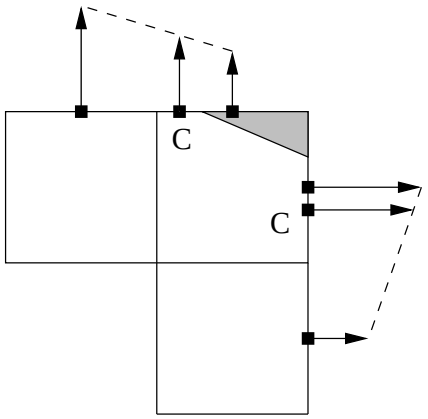

Figure 4.5: The value of v_d or $\mathbf{t}_d$ at the face centroid at point C, is found by linearly interpolating from the two neighboring face-centered values.

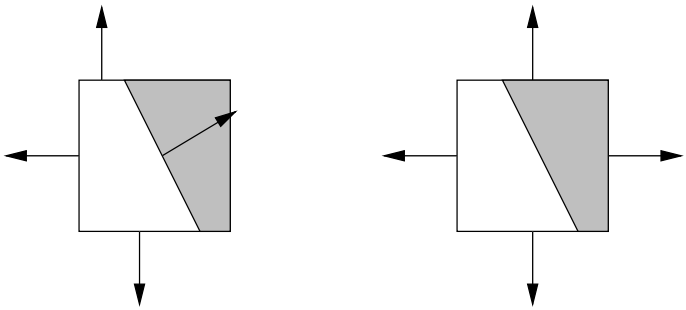

Figure 4.6: (Left) In the conservative divergence stencil, the normal components of $\mathbf{v}$ or $\mathbf{t}$ are needed at face-centroids and the EB centroid. (Right) In the non-conservative divergence stencil, the normal components of $\mathbf{v}$ or $\mathbf{t}$ are needed at face-centers, even if the face is irregular or covered.

face is regular, irregular, or covered, as illustrated in Figure 4.6:

$$(\nabla \cdot \mathbf{v})_{\mathbf{i}}^{NC,FC\to CC} = \frac{1}{h}\sum_{d=0}^{1}\sum_{\pm=+,-} \pm v_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}. \tag{4.45}$$

The non-conservative divergence is equivalent to the four-point divergence (4.4), except that the non-conservative divergence is defined for irregular cells. For regular cells, the conservative and non-conservative divergence discretizations are equivalent. Both discretizations have analogous forms for the divergence of a tensor field.

Embedded boundary methods have been used with success in a wide variety of problems, with a review given in [23]. Recently, new techniques have been developed for solving elliptic equations [42], parabolic equations [54], and systems of hyperbolic conservation laws [24]. This thesis uses solution techniques for each of these classes of PDEs. The stability and accuracy of these methods have been analyzed, with a particular focus on the "small-cell" problem [23], in which $\kappa_{\mathbf{i}} \ll 1$.

Typical elliptic and parabolic equations that are solved are the Poisson equation:

$$\Delta^h \phi = b, \tag{4.46}$$

and the Helmholtz equation:

$$(\mathbf{I} - D\Delta^h)\phi = b, \tag{4.47}$$

where:

$$(\Delta^h \phi)_{\mathbf{i}} = \left[\nabla \cdot (\nabla \phi)^{CC \to FC}\right]_{\mathbf{i}}^{C,FC \to CC}. \tag{4.48}$$

Note that for irregular cells, the EB centroid normal gradient, $(\mathbf{n} \cdot \nabla \phi)^{CC \to EB}$, is also needed to compute the conservative divergence. The discretizations for the face-centered normal gradient, $(\nabla \phi)^{CC \to FC}$, and EB centroid normal gradient are described in Appendix Sections A.2.2 and A.2.3.

For equations (4.46) and (4.47), the potential stability problem coming from the division by $\kappa_{\mathbf{i}}$ in (4.44) is eliminated by multiplying both sides of the equation by $\kappa_{\mathbf{i}}$. It was shown in [42, 54] that solving the scaled versions of (4.46) and (4.47) with a straightforward centered difference calculation of the gradients at centroids leads to a stable method. The choice of gradient discretizations in order to obtain a stable discretization for three-dimensional problems is non-trivial, as is discussed in [71]. For parabolic problems, the use of a Crank-Nicolson discretization has been shown to be unstable, due to the interaction of a non-symmetric matrix representation of the parabolic operator with the neutral stability of Crank-Nicolson [43]. The remedy

proposed in [54] is to choose a more advanced semi-implicit discretization which is stable in the presence of embedded boundaries, such as the method developed by Twizell, et al. [78]. The truncation error for these methods is $\mathcal{O}(h^2)$ for regular cells and $\mathcal{O}(h/\kappa_{\mathbf{i}})$ for irregular cells. The modified equation analysis in [42] indicates that the effect of the truncation error is an additional forcing on the PDE. This forcing acts over a region of size $\mathcal{O}(\kappa_{\mathbf{i}})$, which cancels the volume fraction in the denominator of the truncation error. Thus, the $\mathcal{O}(h)$ forcing of the truncation error in irregular cells acts over a region of width $\mathcal{O}(h)$. The smoothing obtained from applying the solution operator leads to a solution error which is $\mathcal{O}(h^2)$ uniformly for all cells. This phenomena is also supported by numerical evidence, in which second-order solution error convergence is observed in L^∞.

For hyperbolic conservation laws of the form (4.1), due to the explicit nature of the solution discretization, the time step restriction based on using (4.44) to compute the flux divergence is no better than $\Delta t = \mathcal{O}(h\sqrt{\kappa_{\mathbf{i}}}/|\Lambda_{\mathbf{i},\max}|)$ in two dimensions. There have been a number of attempts to deal with this problem, including merging the small control volumes with nearby larger ones [19, 67] and the development of specialized stencils that cancel the required terms [9]. The approach taken here is to expand the range of influence of the small control volumes by using a linear combination of conservative and non-conservative divergence stencils for approximating the fluxes [24] along with a redistribution procedure to maintain global conservation. The truncation error for this method is $\mathcal{O}(h^2)$ for regular cells and $\mathcal{O}(h)$ for irregular cells. Thus, the modified equation analysis predicts second-order solution error convergence in L^1 and first-order in L^∞, as supported by numerical evidence.

4.3 Reconstruction Algorithm

The reconstruction algorithm computes face-centered, time-centered estimates of the state variables which are later used to compute the flux divergence. Following [22, 24], the reconstruction algorithm is divided into a normal predictor step

and a transverse predictor step. In the normal predictor, a one-dimensional Taylor series extrapolation in space and time, along with an upwind procedure in the form of a characteristic projection operator, are used to obtain a one-dimensional MAC state. In the transverse predictor, the transverse derivatives omitted in the normal predictor are accounted for by taking differences of the one-dimensional MAC state, yielding the preliminary MAC state. The algorithm in [22, 24] will be modified to eliminate the propagation of compressive waves inherent to the equations of motion, as well as account for the non-zero source terms. Additionally, the normal and transverse predictor require the use of a Riemann solver and well-posed boundary conditions, which are derived for the equations in this thesis. After the reconstruction step, a MAC projection is performed on the preliminary MAC velocity to enforce the divergence constraint, yielding the final MAC state.

4.3.1 Normal Predictor

The first step in the reconstruction algorithm is the normal predictor, which computes a one-dimensional MAC state using a one-dimensional Taylor series extrapolation in space and time followed by a Riemann solver. The Taylor series extrapolation, illustrated in Figure 4.7, computes two face-centered, time-centered states at each face, which are centered about the two neighboring cell-centers:

$$\mathbf{W}_{\mathbf{i},\pm,d} = \mathbf{W}_{\mathbf{i}}^{n} \pm \frac{h}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial x_d} + \frac{\Delta t}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial t}. \tag{4.49}$$

Consider a one-dimensional version of the primitive variable form:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}_d(\mathbf{W}) \cdot \frac{\partial \mathbf{W}}{\partial x_d} = \mathbf{S}_W, \tag{4.50}$$

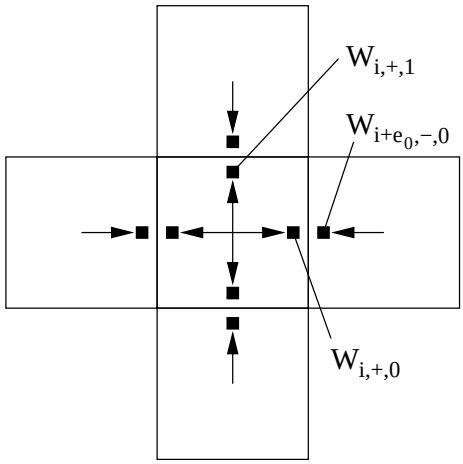

Figure 4.7: In the normal predictor, the face-centered, time-centered states, $\mathbf{W}_{\mathbf{i},\pm,d}$, are computed using Taylor series extrapolations centered about each neighboring cell-center.

The time derivative in (4.49) is replaced with the one-dimensional primitive variable form:

$$\begin{aligned}\mathbf{W}_{\mathbf{i},\pm,d} &= \mathbf{W}_{\mathbf{i}}^n \pm \frac{h}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^n}{\partial x_d} + \frac{\Delta t}{2}\underbrace{\left[-\mathbf{A}_d(\mathbf{W}_{\mathbf{i}}^n)\cdot\frac{\partial \mathbf{W}_{\mathbf{i}}^n}{\partial x_d} + \mathbf{S}_{W,\mathbf{i}}\right]}_{\partial \mathbf{W}_{\mathbf{i}}^n/\partial t} \\ &= \mathbf{W}_{\mathbf{i}}^n + \left[\pm\frac{h}{2}\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_d(\mathbf{W}_{\mathbf{i}}^n)\right]\cdot\frac{\partial \mathbf{W}_{\mathbf{i}}^n}{\partial x_d} + \frac{\Delta t}{2}\mathbf{S}_{W,\mathbf{i}}. \end{aligned} \tag{4.51}$$

The source term vector contains a combination of implicit and explicit terms, so for now the temporal location of these terms, as well as the spatial discretization, is left unspecified. The spatial derivative in (4.51) can be represented by the following fourth-order difference approximation:

$$\Delta_d^{4C}\mathbf{W}_{\mathbf{i}}^n = \frac{1}{12}\mathbf{W}_{\mathbf{i}-2\mathbf{e}_d}^n - \frac{2}{3}\mathbf{W}_{\mathbf{i}-\mathbf{e}_d}^n + \frac{2}{3}\mathbf{W}_{\mathbf{i}+\mathbf{e}_d}^n - \frac{1}{12}\mathbf{W}_{\mathbf{i}+2\mathbf{e}_d}^n; \tag{4.52}$$

$$\frac{\Delta_d^{4C}\mathbf{W}_{\mathbf{i}}^n}{h} = \frac{\partial \mathbf{W}_{\mathbf{i}}^n}{\partial x_d} + \mathcal{O}(h^4). \tag{4.53}$$

However, this finite difference approximation cannot simply replace the spatial derivative in (4.51). Godunov [36] proved that all explicit methods of second-order or greater for hyperbolic problems will develop unphysical oscillations. In order to prevent these oscillations from forming, van Leer [80] introduced a slope limiting procedure where the derivative approximation locally drops order in cells where oscillations would develop if limiting were not used. Also, due to the presence of domain boundaries and EBs, the stencil points for the fourth-order approximation are not necessarily available, requiring the use of one-sided, lower-order difference approximations. The details for the fourth-order derivative approximation with van Leer limiting in an EB framework, denoted $\Delta_d^4 \mathbf{W}_\mathbf{i}$, are given in Appendix Section A.1. Following [24], the spatial derivative in (4.51) will be replaced with $\Delta_d^4 \mathbf{W}_\mathbf{i}^n$, rather than $\Delta_d^{4C} \mathbf{W}_\mathbf{i}^n$.

The difference approximation $\Delta_d^4 \mathbf{W}_\mathbf{i}^n$ can be uniquely decomposed into a linear combination of the eigenvectors of $\mathbf{A}_d(\mathbf{W}_\mathbf{i}^n)$:

$$\Delta_d^4 \mathbf{W}_\mathbf{i}^n = \sum_{k=0}^{N-1} \beta_k \mathbf{r}_k; \qquad \beta_k = \mathbf{l}_k \cdot \Delta_d^4 \mathbf{W}_\mathbf{i}^n, \tag{4.54}$$

where $\mathbf{r}_k$ and $\mathbf{l}_k$ are the corresponding right column eigenvectors and left row eigenvectors of $\mathbf{A}_d(\mathbf{W}_\mathbf{i}^n)$, normalized so that $\mathbf{r}_k \cdot \mathbf{l}_k = 1$. In particular, the matrix of right column eigenvectors of $\mathbf{A}_0(\mathbf{W})$ is given in (4.38), and the matrix of left row

eigenvectors is given by:

$$\mathbf{L}_0 = \begin{bmatrix} -\frac{1}{2\sqrt{2}\alpha_0 c_0} & 0 & -\frac{1}{4\alpha_0^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\tau_{10}}{2\alpha_0 c_0} & -\frac{1}{2c_0} & \frac{\tau_{10}}{2\alpha_0^2} & -\frac{1}{2\alpha_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\tau_{10}^2}{\alpha_0^2} & -\frac{2\tau_{10}}{\alpha_0} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{g_{00}\alpha_0 - g_{01}\tau_{10}}{2\alpha_0^2} & \frac{g_{01}}{\alpha_0} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{g_{10}\alpha_0 - g_{11}\tau_{10}}{2\alpha_0^2} & \frac{g_{11}}{\alpha_0} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{\tau_{10}}{2\alpha_0 c_0} & \frac{1}{2c_0} & \frac{\tau_{10}}{2\alpha_0^2} & -\frac{1}{2\alpha_0} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2\sqrt{2}\alpha_0 c_0} & 0 & -\frac{1}{4\alpha_0^2} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.55}$$

A characteristic projection operator, $P_\pm$, is applied to the difference approximation in the Taylor series extrapolation, which is an upwind technique that discards the components of the differences that do not propagate toward the face. The extrapolation is now given by:

$$\mathbf{W}_{\mathbf{i},\pm,d} = \mathbf{W}_{\mathbf{i}}^n + \frac{1}{2}\left[\pm\mathbf{I} - \frac{\Delta t}{h}\mathbf{A}_d(\mathbf{W}_{\mathbf{i}}^n)\right] \cdot P_\pm(\Delta_d^4 \mathbf{W}_{\mathbf{i}}^n) + \frac{\Delta t}{2}\mathbf{S}_{W,\mathbf{i}}; \tag{4.56}$$

$$P_\pm(\Delta_d^4 \mathbf{W}_{\mathbf{i}}^n) = \sum_{k:\Lambda_k \gtrless 0} \beta_k \mathbf{r}_k, \tag{4.57}$$

where Λ_k are the eigenvalues of $\mathbf{A}_d(\mathbf{W}_{\mathbf{i}}^n)$, given in (4.37). Since $\mathbf{A}_d(\mathbf{W}_{\mathbf{i}}^n)\cdot\mathbf{r}_k = \Lambda_k \mathbf{r}_k$, the extrapolation can be simplified to:

$$\mathbf{W}_{\mathbf{i},\pm,d} = \mathbf{W}_{\mathbf{i}}^n + \frac{1}{2}\sum_{k:\Lambda_k \gtrless 0}\left(\pm 1 - \frac{\Delta t}{h}\Lambda_k\right)\beta_k \mathbf{r}_k + \frac{\Delta t}{2}\mathbf{S}_{W,\mathbf{i}}. \tag{4.58}$$

Variable Partitioning: Unfortunately, the equations of motions contain compressive waves which violate the divergence constraint, as noted in [77]. The longitudinal modes corresponding to $\Lambda_{0,8} = u_0 \pm \sqrt{2}c_0$ are compressive waves, as they

permit a discontinuity in the normal velocity. This can be seen due to the presence of non-zero $\mp\sqrt{2}\alpha_0 c_0$ terms in the u_0 position in $\mathbf{r}_0$ and $\mathbf{r}_8$. A discontinuity in the normal velocity cannot physically exist in a one-dimensional incompressible problem. To remedy this situation, the approach taken is to partition the primitive variables into "active" and "inactive" groups $\mathbf{W}_{A,d} \in \mathbb{R}^{N_A}$ and $\mathbf{W}_{I,d} \in \mathbb{R}^{N_I}$. A similar approach was used in [29] to enforce the divergence-free constraint on the normal magnetic field in the equations of ideal magnetohydrodynamics (MHD). In the normal predictor, only the active variables are subject to a characteristic projection operator, thus eliminating the numerical propagation of compressive waves. The inactive variables are chosen to be the normal velocity u_d and the normal-normal stress τ_{dd}, which is the stress component associated with the compressive waves.

Now, the primitive variables are partitioned and ordered as follows:

$$\begin{aligned} \mathbf{W}_0' &= [\mathbf{W}_{A,0} \mid \mathbf{W}_{I,0}]^T \\ &= [u_1, \tau_{10}, \tau_{11}, g_{00}, g_{10}, g_{01}, g_{11} \mid u_0, \tau_{00}]^T, \end{aligned} \tag{4.59}$$

and the one-dimensional primitive variable form is partitioned as follows:

$$\frac{\partial \mathbf{W}_d'}{\partial t} + \mathbf{A}_d'(\mathbf{W}) \cdot \frac{\partial \mathbf{W}_d'}{\partial x_d} = \mathbf{S}_W', \tag{4.60}$$

$$\frac{\partial}{\partial t}\left[\begin{array}{c} \mathbf{W}_{A,d} \\ \hline \mathbf{W}_{I,d} \end{array}\right] + \left[\begin{array}{c|c} \mathbf{A}_{AA,d}(\mathbf{W}) & \mathbf{A}_{AI,d}(\mathbf{W}) \\ \hline \mathbf{A}_{IA,d}(\mathbf{W}) & \mathbf{A}_{II,d}(\mathbf{W}) \end{array}\right] \cdot \frac{\partial}{\partial x_d}\left[\begin{array}{c} \mathbf{W}_{A,d} \\ \hline \mathbf{W}_{I,d} \end{array}\right] = \left[\begin{array}{c} \mathbf{S}_{W,A} \\ \hline \mathbf{S}_{W,I} \end{array}\right]. \tag{4.61}$$

The primitive coefficient matrix $\mathbf{A}_0'$ is given as:

$$\mathbf{A}_0' = \left[\begin{array}{c|c} \mathbf{A}_{AA,0} & \mathbf{A}_{AI,0} \\ \hline \mathbf{A}_{IA,0} & \mathbf{A}_{II,0} \end{array}\right]$$

$$= \left[\begin{array}{ccccccc|cc} u_0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha_0 & u_0 & 0 & 0 & 0 & 0 & 0 & -\tau_{10} & 0 \\ -2\tau_{10} & 0 & u_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_{01} & 0 & 0 & u_0 & 0 & 0 & 0 & g_{00} & 0 \\ g_{11} & 0 & 0 & 0 & u_0 & 0 & 0 & g_{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & u_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_0 & -\frac{1}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\alpha_0 & u_0 \end{array}\right]. \tag{4.62}$$

The eigenvalues of $\mathbf{A}_{AA,0}$ are:

$$\mathbf{\Lambda}(\mathbf{A}_{AA,0}) = \left\{ u_0 - c_0, \underbrace{u_0, u_0, u_0, u_0, u_0}_{\text{5-fold degenerate}}, u_0 + c_0 \right\}. \tag{4.63}$$

The right column eigenvectors and left row eigenvectors of $\mathbf{A}_{AA,0}$, normalized such that $\mathbf{r}_k \cdot \mathbf{l}_k = 1$, are:

$$\mathbf{R}_{AA,0} = \left[\begin{array}{ccccccc} -c_0 & 0 & 0 & 0 & 0 & 0 & c_0 \\ -\alpha_0 & 0 & 0 & 0 & 0 & 0 & -\alpha_0 \\ -2\tau_{10} & 1 & 0 & 0 & 0 & 0 & -2\tau_{10} \\ g_{01} & 0 & 1 & 0 & 0 & 0 & g_{01} \\ g_{11} & 0 & 0 & 1 & 0 & 0 & g_{11} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right], \tag{4.64}$$

$$\mathbf{L}_{AA,0} = \begin{bmatrix} -\frac{1}{2c_0} & -\frac{1}{2\alpha_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{2\tau_{10}}{\alpha_0} & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{g_{01}}{\alpha_0} & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{g_{11}}{\alpha_0} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2c_0} & -\frac{1}{2\alpha_0} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.65}$$

Thus, the eigenvalues $u_0 \pm \sqrt{2}c_0$ do not exist in the active partition, and the remaining eigenvalues and eigenvectors remain unchanged from the unpartitioned system. For $d = 1$, the primitive variables and advection matrices are ordered as follows:

$$\begin{aligned} \mathbf{W}_1' &= [\mathbf{W}_{A,1} \,|\, \mathbf{W}_{I,1}]^T \\ &= [u_0, \tau_{00}, \tau_{10}, g_{00}, g_{10}, g_{01}, g_{11} \,|\, u_1, \tau_{11}]^T, \end{aligned} \tag{4.66}$$

$$\mathbf{A}_1' = \left[\begin{array}{ccccccc|cc} u_1 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 \\ -2\tau_{10} & u_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha_1 & 0 & u_1 & 0 & 0 & 0 & 0 & -\tau_{10} & 0 \\ 0 & 0 & 0 & u_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u_1 & 0 & 0 & 0 & 0 \\ g_{00} & 0 & 0 & 0 & 0 & u_1 & 0 & g_{01} & 0 \\ g_{10} & 0 & 0 & 0 & 0 & 0 & u_1 & g_{11} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_1 & -\frac{1}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\alpha_1 & u_1 \end{array}\right]. \tag{4.67}$$

The eigenvalues and eigenvectors are:

$$\mathbf{\Lambda}(\mathbf{A}_{AA,1}) = \{u_1 - c_1, \underbrace{u_1, u_1, u_1, u_1, u_1}_{\text{5-fold degenerate}}, u_1 + c_1\}, \tag{4.68}$$

$$\mathbf{R}_{AA,1} = \begin{bmatrix} -c_1 & 0 & 0 & 0 & 0 & 0 & c_1 \\ -2\tau_{10} & 1 & 0 & 0 & 0 & 0 & -2\tau_{10} \\ -\alpha_1 & 0 & 0 & 0 & 0 & 0 & -\alpha_1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ g_{00} & 0 & 0 & 0 & 1 & 0 & g_{00} \\ g_{10} & 0 & 0 & 0 & 0 & 1 & g_{10} \end{bmatrix}, \tag{4.69}$$

$$\mathbf{L}_{AA,1} = \begin{bmatrix} -\frac{1}{2c_1} & 0 & -\frac{1}{2\alpha_1} & 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{2\tau_{10}}{\alpha_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{g_{00}}{\alpha_1} & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{g_{10}}{\alpha_1} & 0 & 0 & 0 & 1 \\ \frac{1}{2c_1} & 0 & -\frac{1}{2\alpha_1} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.70}$$

The Taylor series extrapolation is now partitioned as:

$$\mathbf{W}'_{\mathbf{i},\pm,d} = \mathbf{W}_{\mathbf{i}}^{n'} \pm \frac{h}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n'}}{\partial x_d} + \frac{\Delta t}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n'}}{\partial t}, \tag{4.71}$$

$$\begin{bmatrix} \mathbf{W}_{A,\mathbf{i},\pm,d} \\ \mathbf{W}_{I,\mathbf{i},\pm,d} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{A,\mathbf{i}}^{n} \\ \mathbf{W}_{I,\mathbf{i}}^{n} \end{bmatrix} \pm \frac{h}{2}\frac{\partial}{\partial x_d}\begin{bmatrix} \mathbf{W}_{A,\mathbf{i}}^{n} \\ \mathbf{W}_{I,\mathbf{i}}^{n} \end{bmatrix} + \frac{\Delta t}{2}\frac{\partial}{\partial t}\begin{bmatrix} \mathbf{W}_{A,\mathbf{i}}^{n} \\ \mathbf{W}_{I,\mathbf{i}}^{n} \end{bmatrix}. \tag{4.72}$$

Replacing the partitioned time derivative in (4.72) with the partitioned one-dimensional primitive variable form (4.61) leads to separate equations for the active and inactive variables:

$$\begin{aligned} \mathbf{W}_{A,\mathbf{i},\pm,d} &= \mathbf{W}_{A,\mathbf{i}}^{n} + \left[\pm\frac{h}{2}\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_{AA,d}(\mathbf{W}_{\mathbf{i}}^{n})\right]\frac{\partial \mathbf{W}_{A,\mathbf{i}}^{n}}{\partial x_d} \\ &\quad + \frac{\Delta t}{2}\left[-\mathbf{A}_{AI,d}(\mathbf{W}_{\mathbf{i}}^{n})\frac{\partial \mathbf{W}_{I,\mathbf{i}}^{n}}{\partial x_d} + \mathbf{S}_{W,A,\mathbf{i}}\right], \end{aligned} \tag{4.73}$$

$$\mathbf{W}_{I,\mathbf{i},\pm,d} = \mathbf{W}^n_{I,\mathbf{i}} + \left[\pm\frac{h}{2}\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_{II,d}(\mathbf{W}^n_{\mathbf{i}})\right]\frac{\partial \mathbf{W}^n_{I,\mathbf{i}}}{\partial x_d} + \frac{\Delta t}{2}\left[-\mathbf{A}_{IA,d}(\mathbf{W}^n_{\mathbf{i}})\frac{\partial \mathbf{W}^n_{A,\mathbf{i}}}{\partial x_d} + \mathbf{S}_{W,I,\mathbf{i}}\right]. \quad (4.74)$$

Following [29], for the active variables, the characteristic projection operator is applied to the active differences only, and second-order, non-limited differences (see Appendix Section A.1) are used for the inactive differences:

$$\mathbf{W}_{A,\mathbf{i},\pm,d} = \mathbf{W}^n_{A,\mathbf{i}} + \frac{1}{2}\sum_{k:\Lambda_k \gtrless 0}\left(\pm 1 - \frac{\Delta t}{h}\Lambda_k\right)\beta_k \mathbf{r}_k + \frac{\Delta t}{2}\left[-\frac{1}{h}\mathbf{A}_{AI,d}(\mathbf{W}^n_{\mathbf{i}})\mathcal{D}^2_d \mathbf{W}^n_{I,\mathbf{i}} + \mathbf{S}_{W,A,\mathbf{i}}\right]; \quad (4.75)$$

$$\beta_k = \mathbf{l}_k \cdot \Delta^4_d \mathbf{W}^n_{A,\mathbf{i}}, \quad (4.76)$$

where $\Lambda_k, \mathbf{r}_k$, and $\mathbf{l}_k$ now correspond to the active-active partition $\mathbf{A}_{AA,d}(\mathbf{W}^n_{\mathbf{i}})$ of $\mathbf{A}'_d(\mathbf{W}^n_{\mathbf{i}})$. For the inactive variables, second-order non-limited differences are applied to both the active and inactive differences:

$$\mathbf{W}_{I,\mathbf{i},\pm,d} = \mathbf{W}^n_{I,\mathbf{i}} + \left[\pm\frac{1}{2}\mathbf{I} - \frac{\Delta t}{2h}\mathbf{A}_{II,d}(\mathbf{W}^n_{\mathbf{i}})\right]\cdot \mathcal{D}^2_d \mathbf{W}^n_{I,\mathbf{i}} - \frac{\Delta t}{2h}\mathbf{A}_{IA,d}(\mathbf{W}^n_{\mathbf{i}})\cdot \mathcal{D}^2_d \mathbf{W}^n_{A,\mathbf{i}} + \frac{\Delta t}{2}\mathbf{S}_{W,I,\mathbf{i}}. \quad (4.77)$$

For irregular cells, the Taylor series extrapolation is only performed to open faces, as illustrated in Figure 4.8. An extrapolation procedure will be used to obtain the covered face state. Also, note that there is only one extrapolated state at domain boundaries.

Velocity source terms: The velocity source terms have the following dis-

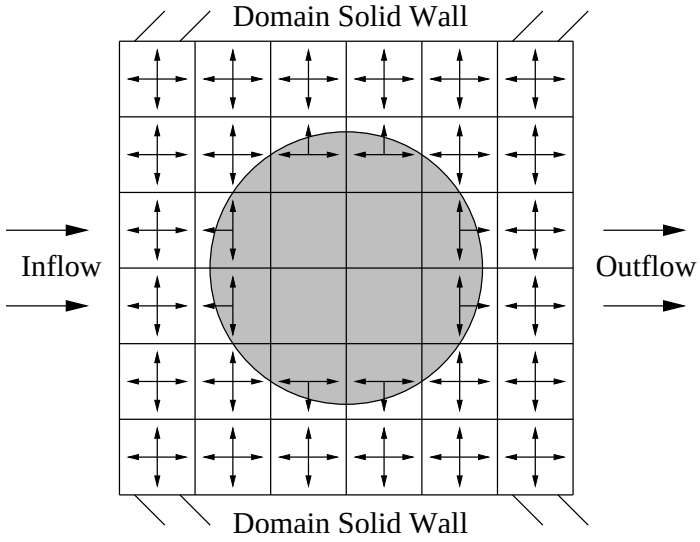

Figure 4.8: For each open cell, a Taylor series extrapolation is performed to obtain a face-centered, time-centered state at each open face.

cretization:

$$\mathbf{u}_{\mathbf{i},\pm,d} = \tilde{\mathbf{u}}_{\mathbf{i},\pm,d} + \frac{\Delta t}{2} \underbrace{\left(-\frac{1}{\rho} \nabla p_{\mathbf{i}}^{n-\frac{1}{2}} + \kappa_{\mathbf{i}} \nu (\Delta^h \mathbf{u}^n)_{\mathbf{i}} \right)}_{\mathbf{S}_{W,\mathbf{u},\mathbf{i}}}, \tag{4.78}$$

where $\tilde{\mathbf{u}}_{\mathbf{i},\pm,d}$ are the velocity terms from the normal predictor Taylor series extrapolation without the effect of the source terms. The time-lagged pressure gradient is known and was previously computed with the pressure-projection method described in Section 4.5.2. The Laplacian is computed component-wise using the stencil $\Delta^h \phi$ from (4.48). For regular cells away from domain boundaries, the Laplacian stencil becomes the standard five-point difference approximation, as illustrated in Figure 4.9:

$$(\Delta^h \phi)_{\mathbf{i}} = \frac{\phi_{\mathbf{i}+\mathbf{e}_0} + \phi_{\mathbf{i}+\mathbf{e}_1} - 4\phi_{\mathbf{i}} + \phi_{\mathbf{i}-\mathbf{e}_0} + \phi_{\mathbf{i}-\mathbf{e}_1}}{h^2}. \tag{4.79}$$

Boundary conditions for the Laplacian are enforced in the normal gradient calculation. The physical boundary conditions for the Laplacian in (4.78) are as follows: a prescribed Dirichlet value is used at inflow ($\mathbf{u} = \mathbf{u}_{\text{in}}$), a homogeneous Dirichlet, or no-slip condition ($\mathbf{u} = 0$) is used at domain and EB solid walls, and a homogeneous

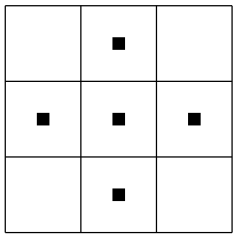

Figure 4.9: Stencil points for the five-point Laplacian for regular cells.

Neumann condition ($\partial \mathbf{u}/\partial x_0 = 0$) is used at outflow. The general Laplacian stencil contains a factor of the inverse volume fraction, $1/\kappa_{\mathbf{i}}$, so the Laplacian in (4.78) is scaled by κ_i to cancel the potentially small volume fraction in the denominator, providing stability for small control volumes at the cost of local accuracy.

Stress source terms: For the stress source terms, a combination of implicit and explicit terms are used, giving the following discretization:

$$\boldsymbol{\tau}_{\mathbf{i},\pm,d} = \tilde{\boldsymbol{\tau}}_{\mathbf{i},\pm,d} + \frac{\Delta t}{2} \underbrace{\left[-\frac{1}{\lambda} \boldsymbol{\tau}_{\mathbf{i},\pm,d} + \left(\frac{\mu_p}{\lambda} - a^2 \right) 2\mathbf{D}_{\mathbf{i}}^n \right]}_{\mathbf{S}_{W,\boldsymbol{\tau},\mathbf{i}}}, \tag{4.80}$$

where $\tilde{\boldsymbol{\tau}}_{\mathbf{i},\pm,d}$ are the stress terms from the normal predictor Taylor series extrapolation without the effect of the source terms. This equation can be rearranged to give:

$$\boldsymbol{\tau}_{\mathbf{i},\pm,d} = \left(\frac{2\lambda}{2\lambda + \Delta t} \right) \tilde{\boldsymbol{\tau}}_{\mathbf{i},\pm,d} + \left(\frac{\Delta t}{2\lambda + \Delta t} \right) \left(\mu_p - \lambda a^2 \right) 2\mathbf{D}_{\mathbf{i}}^n. \tag{4.81}$$

Note that in the Newtonian limit of the algorithm ($\lambda \to 0$), the stress terms have the same form as the Newtonian stress, but scaled by μ_p rather than μ_s:

$$\boldsymbol{\tau}_{\mathbf{i},\pm,d} \to \mu_p 2\mathbf{D}_{\mathbf{i}}^n = \mu_p \left[\nabla \mathbf{u}^n + (\nabla \mathbf{u}^n)^T \right]_{\mathbf{i}} \quad \text{as} \quad \lambda \to 0. \tag{4.82}$$

The gradients in the rate-of-strain tensor are computed component-wise using the

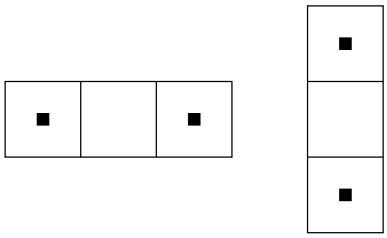

Figure 4.10: Stencil points for the two-point cell-centered gradient for regular cells.

stencil $(\nabla\phi)^{CC\to CC}$, which is the cell-centered gradient of a cell-centered field. For regular cells away from the domain boundary, each component of the gradient is given by the two-point difference approximation, illustrated in Figure 4.10:

$$\nabla\phi_{d,\mathbf{i}}^{CC\to CC} = \frac{\phi_{i+\mathbf{e}_d} - \phi_{\mathbf{i}-\mathbf{e}_d}}{2h}. \tag{4.83}$$

The general discretization for $(\nabla\phi)^{CC\to CC}$ is described in Appendix Section A.2. The physical boundary conditions for the rate-of-strain tensor in (4.81) are as follows: extrapolation boundary conditions are used for the normal velocity at inflow and outflow, and no-flow ($u_n = 0$) is used at solid walls. In this case, no-flow and no-slip conditions are equivalent since the transverse velocity at domain boundaries does not enter into the gradient stencil.

Normal predictor Riemann problem: After the face-centered, time-centered extrapolated states, $\mathbf{W}_{\mathbf{i},\pm,d}$, have been computed, a Riemann solver takes these extrapolated states and computes the one-dimensional MAC state at each open face, $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, as illustrated in Figure 4.11. This process is denoted as:

$$\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \mathcal{R}\left(\mathbf{W}_{\mathbf{i},+,d}, \mathbf{W}_{\mathbf{i}+\mathbf{e}_d,-,d}, d\right). \tag{4.84}$$

The details of the Riemann solver are described in Section 4.3.3. At domain boundaries, there is only one input to the Riemann solver, requiring a special one-sided

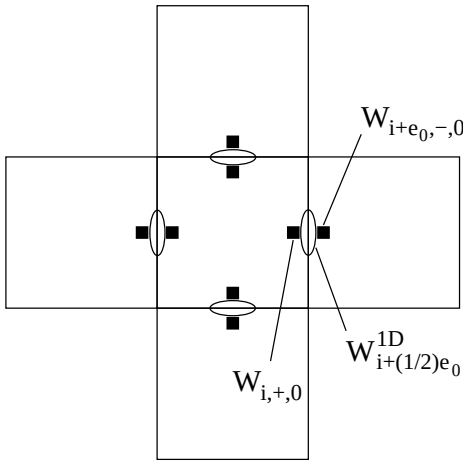

Figure 4.11: A Riemann solver takes the input extrapolated states, $\mathbf{W}_{\mathbf{i},\pm,d}$, and returns the one-dimensional MAC state at each open face, $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$.

Riemann solver, as described in Section 4.3.5. Next, the one-dimensional MAC state at covered faces is computed using the extrapolation procedure described in Appendix Section A.3. At this point, the one-dimensional MAC state is available at each of the four faces in each open cell, regardless of whether the face is regular, irregular, covered, or a domain boundary.

4.3.2 Transverse Predictor

In the transverse predictor, the extrapolated terms from the normal predictor, $\mathbf{W}_{\mathbf{i},\pm,d}$, are updated with transverse derivatives of the one-dimensional MAC state, $\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, to obtain new extrapolated states at open faces, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d}$. This process is illustrated in Figure 4.12 and the details of this procedure are given by:

$$\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d} = \mathbf{W}_{\mathbf{i},\pm,d} - \frac{\Delta t}{2h}\mathbf{A}_{d'}(\mathbf{W}_{\mathbf{i},\pm,d}) \cdot \left(\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \mathbf{W}^{1D}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right); \qquad d \neq d'. \quad (4.85)$$

In the original method, [22, 24], the transverse predictor was posed in terms of

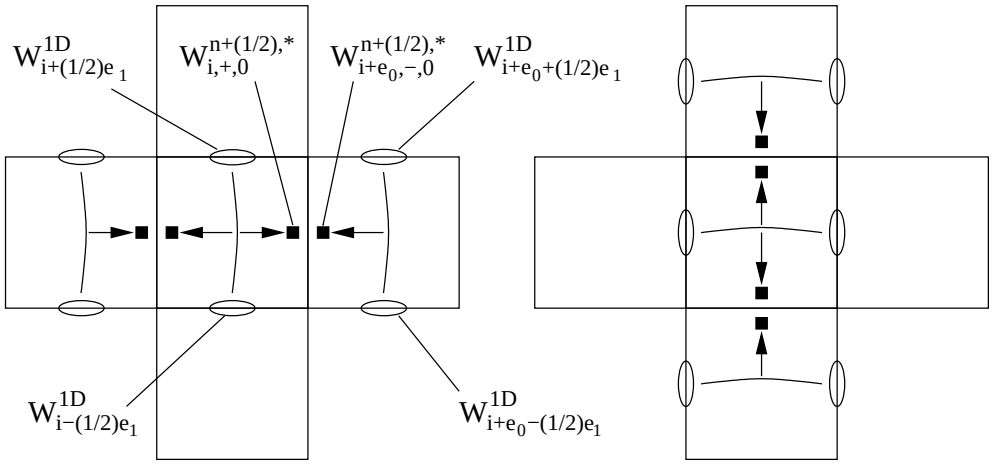

Figure 4.12: In the transverse predictor, the extrapolated terms from the normal predictor, $\mathbf{W}_{\mathbf{i},\pm,d}$, are updated with transverse derivatives of the one-dimensional MAC state, $\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, to obtain new extrapolated states, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d}$.

conserved variables [24]:

$$\mathbf{F}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \mathbf{F}_d(\mathbf{U}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}), \tag{4.86}$$

$$\mathbf{U}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d} = \mathbf{U}_{\mathbf{i},\pm,d} - \frac{\Delta t}{2h}\left(\mathbf{F}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \mathbf{F}^{1D}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right); \qquad d \neq d'. \tag{4.87}$$

However, it has been found that a conservative variable approach affects the convergence rate for the overall algorithm, perhaps due to the coupling of the $\mathbf{g}$ and $\boldsymbol{\tau}$ terms which define $\mathbf{M}$. The primitive variable formulation of the transverse predictor does cause the overall method to suffer from a loss of convergence.

Transverse predictor Riemann problem: Next, a Riemann solver takes the extrapolated states $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d}$ and computes the preliminary MAC state at each open face, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$, as illustrated in Figure 4.13. This process is denoted as:

$$\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \mathcal{R}\left(\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},+,d}, \mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}+\mathbf{e}_d,-,d}, d\right). \tag{4.88}$$

At covered faces, the same extrapolation procedure used in the normal predictor

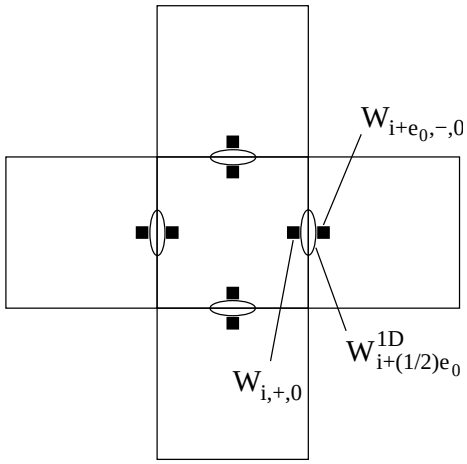

Figure 4.13: A Riemann solver takes the input states, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i},\pm,d}$ and returns the preliminary MAC state at each open face, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$.

is used to obtain the covered face preliminary MAC state, and the same one-sided Riemann solver is used at domain boundaries. At this point, the preliminary MAC state is available at each of the four faces in each open cell, regardless of whether the face is regular, irregular, covered, or a domain boundary. The next step is to perform a MAC projection, as described in Section 4.5.1, to enforce the divergence-free constraint leading to the final MAC state.

4.3.3 Riemann Solver

Both the normal and transverse predictor require the use of a Riemann solver, which takes as inputs a left state, right state, and the direction of the interface normal and determines the future behavior of the system. The solution is a similarity solution, that is, it is a function of $x_d/(t - t_0)$, but the only solution of interest is the future state at the interface where $x_d = 0$. For many systems including gas dynamics [75], MHD [29] and solid mechanics [56, 57], approximate Riemann solvers have been used to avoid the computational expense of the exact solution. For these problems, exact Riemann solvers have been derived [55, 75, 76] primarily

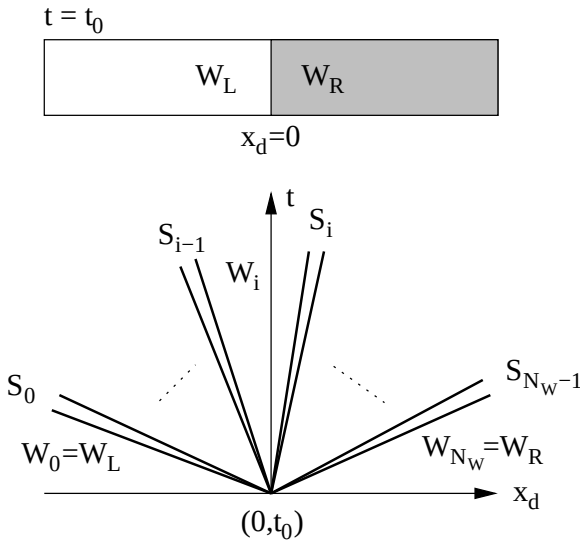

Figure 4.14: A Riemann problem consists of computing the future behavior of a system given two initial states separated by a jump discontinuity. The general solution consists of $N_W + 1$ separate states separated by N_W waves.

to analyze the accuracy of various approximate solvers. Fortunately, the equations of viscoelasticity in this thesis lend themselves to simple, exact solutions that are computationally inexpensive. The Riemann solution is now derived for $d = 0$ with enough generality to solve for the case when $d = 1$.

The Riemann problem will be solved in terms of primitive variables, i.e., given the following system of one-dimensional hyperbolic equations:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}_d(\mathbf{W}) \cdot \frac{\partial \mathbf{W}}{\partial x_d} = \mathbf{S}_W; \qquad \mathbf{W} \in \mathbb{R}^{N_W}, \tag{4.89}$$

with discontinuous initial data of the form:

$$\mathbf{W}(x_d, t_0) = \begin{cases} \mathbf{W}_L, & x_d < 0, \\ \mathbf{W}_R, & x_d > 0. \end{cases} \tag{4.90}$$

The goal is to compute the future behavior of the system . The general solution

consists of $N_W + 1$ separate states separated by N_W waves each with speed S_i in the (x_d, t) plane. The wave speeds are related to, but not necessary equal to the eigenvalues of $\mathbf{A}_d(\mathbf{W})$. If two or more eigenvalues have the same value, the corresponding waves are referred to as degenerate. Each wave emanates from the location of the original discontinuity $(x_d, t) = (0, t_0)$, as illustrated in Figure 4.14.

In general, the solution to the Riemann problem consists of the left state, right state and a series of intermediate states separated by waves associated with the eigenvalues of the primitive coefficient matrix. There is a separate treatment for the active and inactive variables since the same incompressibility-violating waves that were removed in the predictor step cannot exist in the Riemann solution [29]. The solution for the inactive variables for all $(x_d, t); t > t_0$, is simply the average of the left and right states, and will also be used as parameters for the active variable solution.

To derive the active variable solution, the first step is to determine the type of each wave (shock, rarefaction or linearly degenerate) in order to understand how the state variables change across them. Consider the eigenvalues (4.63) and eigenvectors (4.65) of $A_{AA,0}$. It can be illustrated that each of the waves are linearly degenerate, i.e.:

$$\nabla_{\mathbf{W}_{A,0}} \Lambda_k \cdot \mathbf{r}_k = 0; \qquad k = 0, \cdots, 6, \tag{4.91}$$

$$\nabla_{\mathbf{W}_{A,0}} = \left(\frac{\partial}{\partial u_1}, \frac{\partial}{\partial \tau_{10}}, \frac{\partial}{\partial \tau_{11}}, \frac{\partial}{\partial g_{00}}, \frac{\partial}{\partial g_{10}}, \frac{\partial}{\partial g_{01}}, \frac{\partial}{\partial g_{11}} \right)^T. \tag{4.92}$$

Equivalently, a wave is linearly degenerate if the eigenvalue has the same value on each side of the wave. Here, since the eigenvalues are constant with respect to the active variables, (4.91) is trivially true for each wave, and therefore all the waves are linearly degenerate.

A linearly degenerate wave separates two states by a jump discontinuity propagating at speed equal to the eigenvalue [75]. Graphically, such waves are represented

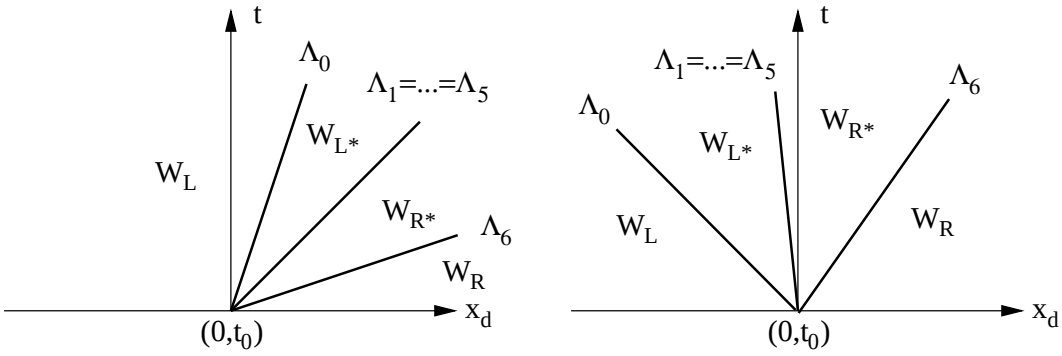

Figure 4.15: Example solution to the Riemann problem; (Left) solution at $x_d/t = 0$ is $\mathbf{W}_L$, (Right) solution at $x_d/t = 0$ is $\mathbf{W}_{R*}$.

in the (x_d, t) plane as passing through the original point of discontinuity with inverse slope equal to the corresponding eigenvalue. Therefore, the complete solution to the Riemann problem consists of 4 states: $\mathbf{W}_L, \mathbf{W}_{L*}, \mathbf{W}_{R*}$, and $\mathbf{W}_R$. The left wave corresponding to $\Lambda_0 = u_0 - c_0$ separates states $\mathbf{W}_L$ and $\mathbf{W}_{L*}$. The five-fold degenerate center wave corresponding to $\Lambda_1 = \cdots = \Lambda_5 = u_0$, which shall be referred to as the contact discontinuity, separates the star states $\mathbf{W}_{L*}$ and $\mathbf{W}_{R*}$. The right wave corresponding to $\Lambda_6 = u_0 + c_0$ separates states $\mathbf{W}_{R}*$ and $\mathbf{W}_R$. Figure 4.15 illustrates example cases when the solution for $x_d/t = 0$ is $\mathbf{W}_L$ and $\mathbf{W}_{R*}$ respectively.

The relative jumps of the active variables across each wave can be determined using the Generalized Riemann Invariants [75]. For a linearly degenerate wave associated with eigenvalue Λ_k, the following N_A ordinary differential equations are true:

$$\frac{\partial W^{(0)}}{r_k^{(0)}} = \frac{\partial W^{(1)}}{r_k^{(1)}} = \cdots = \frac{\partial W^{(N_A-1)}}{r_k^{(N_A-1)}}, \tag{4.93}$$

where $W^{(i)}$ is the i^{th} element in $\mathbf{W}_{A,0}$ and $r_k^{(i)}$ is the i^{th} element in the k^{th} column eigenvector of $\mathbf{R}_{AA,0}$. For the left wave associated with $\Lambda_0 = u_0 - c_0$:

$$\frac{\partial u_1}{-c_0} = \frac{\partial \tau_{10}}{-\alpha_0} = \frac{\partial \tau_{11}}{-2\tau_{10}} = \frac{\partial g_{00}}{g_{01}} = \frac{\partial g_{10}}{g_{11}} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{0}. \tag{4.94}$$

For the center contact discontinuity associated with $\Lambda_1 = \cdots = \Lambda_5 = u_0$:

$$\frac{\partial u_1}{0} = \frac{\partial \tau_{10}}{0} = \frac{\partial \tau_{11}}{1} = \frac{\partial g_{00}}{0} = \frac{\partial g_{10}}{0} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{0}, \tag{4.95}$$

$$\frac{\partial u_1}{0} = \frac{\partial \tau_{10}}{0} = \frac{\partial \tau_{11}}{0} = \frac{\partial g_{00}}{1} = \frac{\partial g_{10}}{0} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{0}, \tag{4.96}$$

$$\frac{\partial u_1}{0} = \frac{\partial \tau_{10}}{0} = \frac{\partial \tau_{11}}{0} = \frac{\partial g_{00}}{0} = \frac{\partial g_{10}}{1} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{0}, \tag{4.97}$$

$$\frac{\partial u_1}{0} = \frac{\partial \tau_{10}}{0} = \frac{\partial \tau_{11}}{0} = \frac{\partial g_{00}}{0} = \frac{\partial g_{10}}{0} = \frac{\partial g_{01}}{1} = \frac{\partial g_{11}}{0}, \tag{4.98}$$

$$\frac{\partial u_1}{0} = \frac{\partial \tau_{10}}{0} = \frac{\partial \tau_{11}}{0} = \frac{\partial g_{00}}{0} = \frac{\partial g_{10}}{0} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{1}. \tag{4.99}$$

For the right wave associated with $\Lambda_6 = u_0 + c_0$:

$$\frac{\partial u_1}{c_0} = \frac{\partial \tau_{10}}{-\alpha_0} = \frac{\partial \tau_{11}}{-2\tau_{10}} = \frac{\partial g_{00}}{g_{01}} = \frac{\partial g_{10}}{g_{11}} = \frac{\partial g_{01}}{0} = \frac{\partial g_{11}}{0}. \tag{4.100}$$

The first important piece of information obtained from (4.95)-(4.99) is that due to the zero in the denominator of the terms containing ∂u_1 and $\partial \tau_{10}$, u_1 and τ_{10} must be constant across the contact discontinuity. Also, for each term in (4.94) and (4.100) except for the term containing $\partial \tau_{11}$, the denominator of each term is a constant in a sense that the value is the same on each side of the corresponding wave. This means that the change in each variable across the wave (excluding τ_{11}), is equal to a constant wave-strength multiplied by the denominator of the corresponding variable. In other words, the complete star states for the active variables excluding τ_{11}, (indicated with a hat above the vector) are:

$$\begin{aligned} \hat{\mathbf{W}}_{A,L*} &= \hat{\mathbf{W}}_{A,L} + \Psi_L \hat{\mathbf{r}}_0 \left(\mathbf{W}_L \right), \\ \hat{\mathbf{W}}_{A,R*} &= \hat{\mathbf{W}}_{A,R} + \Psi_R \hat{\mathbf{r}}_5 \left(\mathbf{W}_R \right). \end{aligned} \tag{4.101}$$

Using a method similar to [76] for ideal MHD, a unique consistent mapping can be made across the states. In particular, u_1 and τ_{10} are constant across the center

contact waves, so these constraints will be used to determine the two wave strengths Ψ_L and Ψ_R. Using (4.93) and (4.94), the following pair of equations must be true:

$$\begin{aligned} u_{1,L*} &= u_{1,L} - \Psi_L c_0, \\ \tau_{10,L*} &= \tau_{10,L} - \Psi_L \alpha_0. \end{aligned} \tag{4.102}$$

Using (4.93) and (4.100), the following equations must be true:

$$\begin{aligned} u_{1,R*} &= u_{1,R} + \Psi_R c_0, \\ \tau_{10,R*} &= \tau_{10,R} - \Psi_R \alpha_0. \end{aligned} \tag{4.103}$$

Since $u_{1,L*} = u_{1,R*}$ and $\tau_{10,L*} = \tau_{10,R*}$ these 4 equations can be reduced to:

$$\begin{bmatrix} c_0 & c_0 \\ \alpha_0 & -\alpha_0 \end{bmatrix} \begin{bmatrix} \Psi_L \\ \Psi_R \end{bmatrix} = \begin{bmatrix} u_{1,L} - u_{1,R} \\ \tau_{10,L} - \tau_{10,R} \end{bmatrix}. \tag{4.104}$$

The unknowns are found by solving this system, and are given by:

$$\begin{bmatrix} \Psi_L \\ \Psi_R \end{bmatrix} = \begin{bmatrix} \frac{\alpha_0\left(u_{1,L}-u_{1,R}\right)+c_0\left(\tau_{10,L}-\tau_{10,R}\right)}{2\alpha_0 c_0} \\ \frac{\alpha_0\left(u_{1,L}-u_{1,R}\right)-c_0\left(\tau_{10,L}-\tau_{10,R}\right)}{2\alpha_0 c_0} \end{bmatrix}. \tag{4.105}$$

To get the $L*$ state for τ_{11}, examine a subset of (4.94):

$$\frac{\partial \tau_{10}}{-\alpha_0} = \frac{\partial \tau_{11}}{-2\tau_{10}} \quad \rightarrow \quad \partial \tau_{11} = \frac{2}{\alpha_0} \tau_{10} \partial \tau_{10}. \tag{4.106}$$

Integrate both sides of the equation across the left wave to get:

$$\int_{\tau_{11,L}}^{\tau_{11,L*}} \partial \tau_{11} = \frac{2}{\alpha_0} \int_{\tau_{10,L}}^{\tau_{10,L*}} \tau_{10} \partial \tau_{10} \quad \rightarrow \quad \tau_{11,L*} = \tau_{11,L} + \frac{1}{\alpha_0} \left(\tau_{10,L*}^2 - \tau_{10,L}^2 \right). \tag{4.107}$$

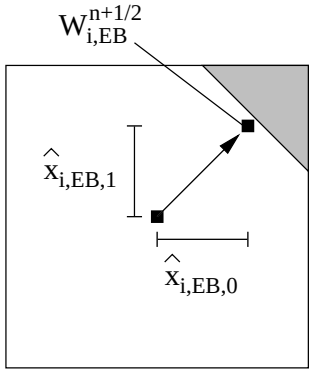

Figure 4.16: A full-dimensional Taylor series extrapolation in space and time is used to compute the EB centroid state, $\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}}$

Similarly, for the right wave:

$$\int_{\tau_{11,R}}^{\tau_{11,R*}} \partial\tau_{11} = \frac{2}{\alpha_0}\int_{\tau_{10,R}}^{\tau_{10,R*}} \tau_{10}\partial\tau_{10} \;\;\rightarrow\;\; \tau_{11,R*} = \tau_{11,R} + \frac{1}{\alpha_0}\left(\tau_{10,R*}^2 - \tau_{10,R}^2\right). \quad (4.108)$$

4.3.4 EB Centroid State Calculation

Since the divergence term in (4.3) is approximated as a linear combination of normal fluxes at the control volume faces, the state variables at the EB centroid must be known in irregular cells. To obtain the time-centered state at the EB centroid $\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}}$, a full-dimensional Taylor extrapolation in space in time is used, as illustrated in Figure 4.16:

$$\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \mathbf{W}_{\mathbf{i}}^{n} + h\hat{x}_{\mathbf{i},EB,0}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial x_0} + h\hat{x}_{\mathbf{i},EB,1}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial x_1} + \frac{\Delta t}{2}\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial t}. \quad (4.109)$$

Replacing the time derivative in this equation with the full-dimensional primitive variable form (2.28) gives the extrapolation:

$$\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \mathbf{W}_{\mathbf{i}}^{n} + \left(h\hat{x}_{\mathbf{i},EB,0}\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_0\right)\cdot\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial x_0} + \left(h\hat{x}_{\mathbf{i},EB,1}\mathbf{I} - \frac{\Delta t}{2}\mathbf{A}_1\right)\cdot\frac{\partial \mathbf{W}_{\mathbf{i}}^{n}}{\partial x_1} + \frac{\Delta t}{2}\mathbf{S}_{W,\mathbf{i}}. \quad (4.110)$$

The choice is made to replace the spatial derivatives with second-order non-limited differences:

$$\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \mathbf{W}_{\mathbf{i}}^{n} + \left(\hat{x}_{\mathbf{i},EB,0}\mathbf{I} - \frac{\Delta t}{2h}\mathbf{A}_0\right)\cdot\mathcal{D}_0^2\mathbf{W}_{\mathbf{i}}^{n} + \left(\hat{x}_{\mathbf{i},EB,1}\mathbf{I} - \frac{\Delta t}{2h}\mathbf{A}_1\right)\cdot\mathcal{D}_1^2\mathbf{W}_{\mathbf{i}}^{n} + \frac{\Delta t}{2}\mathbf{S}_{W,\mathbf{i}}. \tag{4.111}$$

The source terms are evaluated using the same discretization in the normal predictor as given in (4.78) and (4.81). A one-sided Riemann problem must be solved to obtain the solution state at the EB centroid. In order to do this, a rotation is performed on $\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}}$ to a coordinate system where the EB has a normal that points in the $+x_0$ direction. This is done by introducing a rotation matrix using the outward normal:

$$\boldsymbol{\mathcal{R}} = \begin{bmatrix} -n_0 & -n_1 \\ n_1 & -n_0 \end{bmatrix}. \tag{4.112}$$

To rotate the variables, the following is used:

$$\mathbf{u}_{\text{rot}} = \boldsymbol{\mathcal{R}}\cdot\mathbf{u}_{\mathbf{i},EB}^{n+\frac{1}{2}}, \qquad \boldsymbol{\tau}_{\text{rot}} = \boldsymbol{\mathcal{R}}\cdot\boldsymbol{\tau}_{\mathbf{i},EB}^{n+\frac{1}{2}}\cdot\boldsymbol{\mathcal{R}}^T, \qquad \mathbf{g}_{\text{rot}} = \boldsymbol{\mathcal{R}}\cdot\mathbf{g}_{\mathbf{i},EB}^{n+\frac{1}{2}}\cdot\boldsymbol{\mathcal{R}}^T. \tag{4.113}$$

The one-sided Riemann solver for solid walls described in Section 4.3.5 is used, and then the solution state is rotated back to the original frame using:

$$\mathbf{u}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \boldsymbol{\mathcal{R}}^T\cdot\mathbf{u}_{\text{rot}}, \qquad \boldsymbol{\tau}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \boldsymbol{\mathcal{R}}^T\cdot\boldsymbol{\tau}_{\text{rot}}\cdot\boldsymbol{\mathcal{R}}, \qquad \mathbf{g}_{\mathbf{i},EB}^{n+\frac{1}{2}} = \boldsymbol{\mathcal{R}}^T\cdot\mathbf{g}_{\text{rot}}\cdot\boldsymbol{\mathcal{R}}. \tag{4.114}$$

This process is illustrated in Figure 4.17.

4.3.5 One-Sided Riemann Solvers

In the reconstruction algorithm, there are several cases where a one-sided Riemann problem must be solved. This occurs in the normal and transverse predictor at domain boundaries and in the calculation of the time-centered EB centroid state (see

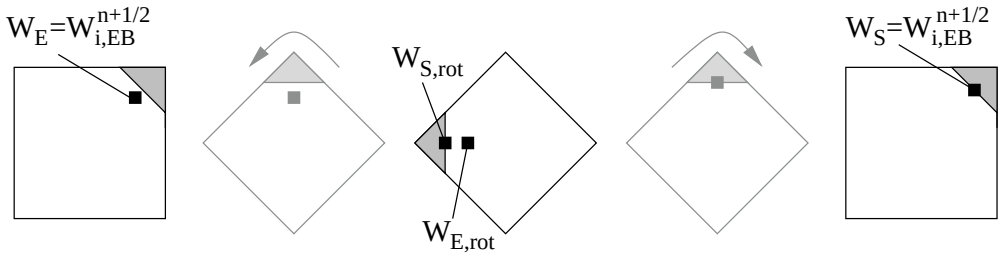

Figure 4.17: In order to solve a one-sided Riemann problem at an EB centroid, the EB centroid state $\mathbf{W}_{\mathbf{i},EB}^{n+\frac{1}{2}}$ is rotated into an orientation where the normal points in the $+x_0$ direction.

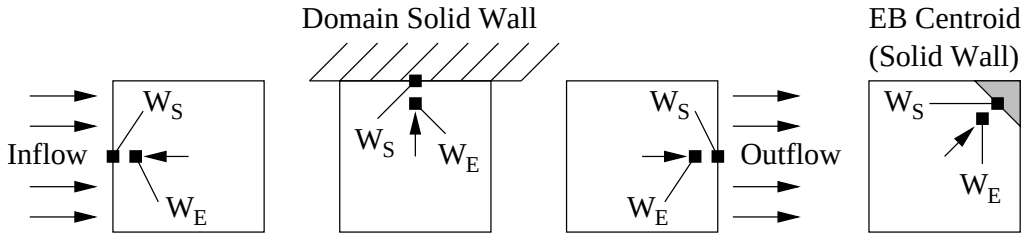

Figure 4.18: There are four cases where a one-sided Riemann solver is used to obtain the solution state $\mathbf{W}_S$: inflow, domain solid walls, outflow, and EB centroids.

Figure 4.18). In each of these cases, an extrapolated state $\mathbf{W}_E$ has been computed, and the actual solution state $\mathbf{W}_S$ is to be determined. The approach taken is to use the methods in [32] for determining well-posed boundary conditions for multi-dimensional finite volume methods for hyperbolic problems. In this method, the eigenvalues and eigenvectors of the coefficient matrices along with physical boundary conditions are used to determine a well-posed solution state. Based on the sign of the eigenvalues at the boundary, $\mathbf{W}_S$ is composed of contributions from either $\mathbf{W}_E$ or physical boundary conditions.

Consider a domain boundary that is on the low side of a Cartesian cell. Positive eigenvalues are said to "point" inside the domain, while negative eigenvalues "point" outside the domain. The reverse is true for domain boundaries that lie on the high side of a cell. For each eigenvalue that points inside the domain, one of the variables in $\mathbf{W}_S$ must be specified to a Dirichlet value based on a physical condition. For each

eigenvalue that points out of the domain, a characteristic condition must be met. Specifically, if Λ_k points out of the domain, a sufficient characteristic condition is given by:

$$\mathbf{l}_k(\mathbf{W}_E) \cdot (\mathbf{W}_E - \mathbf{W}_S) = 0. \tag{4.115}$$

A third possible case is when $\Lambda_k = 0$. This is an ambiguous case, with no well-defined systematic treatment. Here, for each zero eigenvalue, the choice is made to set one element in $\mathbf{W}_S$ to be equal to $\mathbf{W}_E$ as long as any conditions imposed by the non-zero eigenvalues are not violated. By following these rules, there are always exactly the number of conditions to match the number of unknowns in $\mathbf{W}_S$. Note that if all the eigenvalues at a boundary point into the domain, then $\mathbf{W}_S$ must be completely specified using physical conditions. If all the eigenvalues at a boundary point out of the domain, then $\mathbf{W}_S = \mathbf{W}_E$. The one-sided Riemann problem is used for the active variables only, allowing the freedom to choose either a Dirichlet or extrapolated condition for the inactive variables. Specific details are now described for this problem.

For domain boundary solid walls, which will always lie on the high or low side in the x_1 direction, the inactive variables are specified using a no-flow condition on the normal velocity $\mathbf{u}_{1,S} = 0$ and extrapolation on the normal-normal stress $\tau_{11,S} = \tau_{nn,E}$. As for the active variables, it is always true that exactly one eigenvalue points into the domain, one points out of the domain, and the rest are equal to zero. The prescribed physical condition is chosen to be no-slip, $u_t = 0$. Therefore, the shear stress is found simply by applying $\mathbf{l}_k(\mathbf{W}_E) \cdot (\mathbf{W}_E - \mathbf{W}_S) = 0$, where $k = 0$ if the wall normal points in the positive direction and $k = 6$ if the normal points in the negative direction. The transverse velocity and the shear stress are the only two non-zero entries in $\mathbf{l}_0$ and $\mathbf{l}_6$, so all the other active variables are set to the extrapolated state $\mathbf{W}_E$.

At inflow, which lies on the low side of the domain in the x_0 direction, the inactive

variables u_0 and τ_{00} are specified using the prescribed inflow condition. Then, it must be determined whether the inflow is subsonic or supersonic by computing the left wave speed $u_0 - c_0$ using the prescribed inflow. If the flow is supersonic ($u_0 - c_0 > 0$), then all the eigenvalues point inside the domain and $\mathbf{W}_S$ is set to the prescribed inflow. If the flow is subsonic, a characteristic boundary condition is used. Since the fluid velocity is positive, exactly six eigenvalues point into the domain and one eigenvalue points out of the domain. Therefore, $\mathbf{W}_S$ can be specified as equal to the inflow condition for six of the seven active variables. The excluded active variable is the shear stress, which is computed with the equation $\mathbf{l}_0(\mathbf{W}_E) \cdot (\mathbf{W}_E - \mathbf{W}_S) = 0$.

At outflow, which lies on the high side of the domain in the x_0 direction, the inactive variables u_0 and τ_{00} are set to the extrapolated values. Then, it must be determined whether the outflow is subsonic or supersonic by calculating the left wave speed $u_0 - c_0$ using the extrapolated state $\mathbf{W}_E$. If the flow is supersonic ($u_0 - c_0 > 0$), then all the eigenvalues point outside the domain and $\mathbf{W}_S = \mathbf{W}_E$. If the flow is subsonic, a characteristic condition must be used. The fluid velocity is assumed to be positive and therefore exactly one of the seven eigenvalues points into the domain. Therefore, the solution $\mathbf{W}_S$ can only be specified for 1 variable and use characteristic conditions for $\mathbf{l}_k(\mathbf{W}_E) \cdot (\mathbf{W}_E - \mathbf{W}_S) = 0$, $k = 1, \cdots, 6$ to determine the other 6. The domain is assumed to be long enough such that there is no transverse velocity at outflow and the condition $u_1 = 0$ is applied.

4.4 Cell Update Step

Following [24], the control volumes are updated conservatively using fluxes computed with the final MAC state. Afterwards, the conservative source terms are accounted for, followed by a cell-centered projection to enforce the divergence-free constraint on the velocity field as well as update the pressure. Additional steps must be taken to account for the conservative source terms in a manner which preserves a second-order accurate algorithm while capturing the Newtonian and elastic limits

of the Oldroyd-B fluid.

4.4.1 Conservative Update

At this point, the MAC state and the time-centered EB centroid state have been computed, and therefore the corresponding fluxes are available as well. Recall the time-stepping strategy (4.2):

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \left[-\nabla \cdot \mathbf{F} + \mathbf{S}_U \right]. \tag{4.116}$$

For regular cells, the divergence can be represented as the four-point difference approximation in (4.4). For the general case, it would be desirable to use the conservative divergence discretization (4.44), as it leads to local conservation of each conserved variable:

$$\mathbf{U}_{\mathbf{i}}^{n+1} = \mathbf{U}_{\mathbf{i}}^n - \Delta t \left[(\nabla \cdot \mathbf{F})_{\mathbf{i}}^{C,FC\to CC} \right] + \Delta t \mathbf{S}_{U,\mathbf{i}}. \tag{4.117}$$

Unfortunately, the use of the conservative divergence to discretize the divergence term is unstable due to the potentially small volume fraction, $\kappa_{\mathbf{i}}$, in the denominator of the conservative divergence. This is the well-known small-cell problem for EB methods [24]. To resolve this problem, a linear combination of conservative and non-conservative (4.45) divergence discretizations to account for the fluxes is used:

$$\mathbf{U}_{\mathbf{i}}^{n+1} = \mathbf{U}_{\mathbf{i}}^n - \Delta t \left[\kappa_{\mathbf{i}} (\nabla \cdot \mathbf{F})_{\mathbf{i}}^{C,FC\to CC} + (1-\kappa_{\mathbf{i}}) (\nabla \cdot \mathbf{F})_{\mathbf{i}}^{NC,FC\to NC} \right] + \Delta t \mathbf{S}_{U,\mathbf{i}}. \tag{4.118}$$

Since local conservation is not maintained under this procedure, a redistribution procedure is used in order to maintain global conservation. The quantity $\boldsymbol{\delta}_{\mathbf{i}}$ is the difference between the conserved variable increment $\kappa_{\mathbf{i}}(\mathbf{U}_{\mathbf{i}}^{n+1} - \mathbf{U}_{\mathbf{i}}^n)$ given by (4.118)

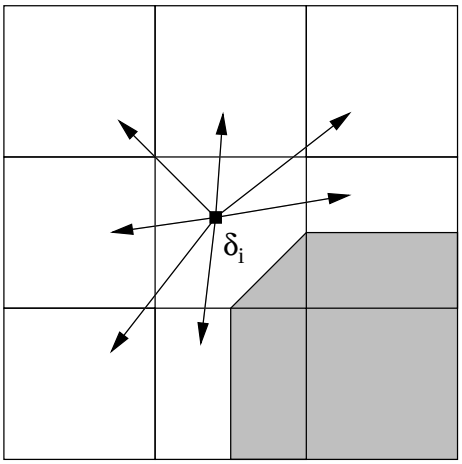

Figure 4.19: The conserved variable vector $\boldsymbol{\delta}_\mathbf{i}$ is distributed into the (up to) nine cells adjacent and including $\mathbf{i}$ in a volume weighted manner.

and a purely conservative update (4.117), ignoring the source terms:

$$\begin{aligned} \boldsymbol{\delta}_\mathbf{i} &= \kappa_\mathbf{i}(\mathbf{U}_\mathbf{i}^{n+1} - \mathbf{U}_\mathbf{i}^{n}) \\ &= \kappa_\mathbf{i}(\kappa_\mathbf{i} - 1)\left[(\nabla \cdot \mathbf{F})_\mathbf{i}^{C,FC\to CC} - (\nabla \cdot \mathbf{F})_\mathbf{i}^{NC,FC\to CC}\right]. \end{aligned} \tag{4.119}$$

A redistribution step is performed, where $\boldsymbol{\delta}_\mathbf{i}$ is distributed into the nine cells adjacent to and including $\mathbf{i}$ in a volume weighted manner (see Figure 4.19), thus preserving global conservation. Note that in regular cells, the conservative and non-conservative divergence discretizations are both the four-point divergence approximation from equation (4.4) and therefore, $\boldsymbol{\delta}_\mathbf{i} = 0$.

4.4.2 Conservative Source Terms

Velocity source terms: After the conservative update and redistribution step, the state variables are updated to account for the conservative source terms. Recall from (4.33) that:

$$\mathbf{S}_{U,\mathbf{u}} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u}. \tag{4.120}$$

The evolutionary part of the incompressible Navier-Stokes equations can be rearranged into the following parabolic diffusion equation:

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \Delta \mathbf{u} + \left[-\nabla \cdot \left(\mathbf{u}\mathbf{u} - \frac{1}{\rho}\boldsymbol{\tau} \right) - \frac{1}{\rho}\nabla p \right]. \tag{4.121}$$

One possible discretization of this equation is:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\Delta t} = \nu \Delta \mathbf{u} + \left[-\nabla \cdot \left(\mathbf{u}\mathbf{u} - \frac{1}{\rho}\boldsymbol{\tau} \right)^{n+\frac{1}{2}} - \frac{1}{\rho}\nabla p^{n-\frac{1}{2}} \right]. \tag{4.122}$$

Standard choices for the temporal discretization of the Laplacian are explicit, $\Delta \mathbf{u}^{n}$, implicit, $\Delta \mathbf{u}^{n+1}$, or semi-implicit, such as Crank-Nicolson, $(\Delta \mathbf{u}^{n} + \Delta \mathbf{u}^{n+1})/2$. In [8], it was illustrated that a Crank-Nicolson discretization of the Laplacian in the absence of EBs would still lead to an overall second-order method, despite the use of the time-lagged pressure gradient. The approach taken in this thesis is to use a more advanced second-order, semi-implicit discretization of the Laplacian that is stable in the presence of embedded boundaries.

The form of the diffusion equation is now modified to include the ideas in [77], in which diffusion coefficient respects the Newtonian and elastic limits of the equation. In particular, in the Newtonian limit ($\lambda \to 0$), the diffusion equation must take the form:

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\mu_s + \mu_p}{\rho} \Delta \mathbf{u} + \left[-\nabla \cdot (\mathbf{u}\mathbf{u})^{n+\frac{1}{2}} - \frac{1}{\rho}\nabla p^{n-\frac{1}{2}} \right], \tag{4.123}$$

whereas in the elastic limit ($\lambda \to \infty$, μ_p/λ finite), the diffusion equation must take the form:

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\mu_s}{\rho} \Delta \mathbf{u} + \left[-\nabla \cdot \left(\mathbf{u}\mathbf{u} - \frac{1}{\rho}\boldsymbol{\tau} \right)^{n+\frac{1}{2}} - \frac{1}{\rho}\nabla p^{n-\frac{1}{2}} \right]. \tag{4.124}$$

In order to derive an equation which respects these limits, an additional modification to the transverse predictor step is required. Recall the transverse predictor step

(4.85):

$$\mathbf{W}_{\mathbf{i},\pm,d}^{n+\frac{1}{2},*} = \mathbf{W}_{\mathbf{i},\pm,d} - \frac{\Delta t}{2h}\mathbf{A}_{d'}(\mathbf{W}_{\mathbf{i},\pm,d}) \cdot \left(\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \mathbf{W}^{1D}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right); \qquad d \neq d'. \quad (4.125)$$

This equation is replaced with:

$$\mathbf{W}_{\mathbf{i},\pm,d}^{n+\frac{1}{2},*} = \tilde{\mathbf{W}}_{\mathbf{i},\pm,d} - \frac{\Delta t}{2h}\mathbf{A}_{d'}(\tilde{\mathbf{W}}_{\mathbf{i},\pm,d}) \cdot \left(\mathbf{W}^{1D}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \mathbf{W}^{1D}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right); \qquad d \neq d'. \quad (4.126)$$

where $\tilde{\mathbf{W}}_{\mathbf{i},\pm,d}$ are the one-dimensional MAC states computed with the assumption that $\mathbf{S}_{W,\boldsymbol{\tau}} = 0$. Note that the differenced terms $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ are one-dimensional MAC states that still do include the effects of $\mathbf{S}_{W,\boldsymbol{\tau}}$. After solving the transverse predictor Riemann problem, the stress terms in the final MAC state is denoted as $\tilde{\boldsymbol{\tau}}^{n+\frac{1}{2}}$ to indicate that the stress source terms in the normal predictor were omitted. Now the stress source terms are accounted for using the semi-implicit equation:

$$\begin{aligned} \boldsymbol{\tau}^{n+\frac{1}{2}} &= \tilde{\boldsymbol{\tau}}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\left[-\frac{1}{\lambda}\boldsymbol{\tau}^{n+\frac{1}{2}} + \left(\frac{\mu_p}{\lambda} - a^2\right) 2\mathbf{D}\right] \\ &= \frac{2}{2\lambda + \Delta t}\left[\lambda\tilde{\boldsymbol{\tau}}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\left(\mu_p - \lambda a^2\right) 2\mathbf{D}\right]. \end{aligned} \quad (4.127)$$

The temporal location and spatial discretization of the rate-of-strain tensor in (4.127) are left unspecified for now. The update equation for velocity can be written as:

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{\mathbf{u}^{n+1,*} - \mathbf{u}^n}{\Delta t} = \left[-\nabla \cdot (\mathbf{u}\mathbf{u} + \frac{1}{\rho}\boldsymbol{\tau})^{n+\frac{1}{2}}\right] + \left[-\frac{1}{\rho}\nabla p^{n-\frac{1}{2}} + \nu\Delta\mathbf{u}\right]. \quad (4.128)$$

Substituting (4.127) into (4.128) and rearranging terms gives a diffusion equation

for the preliminary cell-centered velocity:

$$\begin{aligned}\frac{\mathbf{u}^{n+1,*}-\mathbf{u}^{n}}{\Delta t} &= \frac{1}{\rho}\left[\mu_s+\frac{\left(\mu_p-\lambda a^2\right)\Delta t}{2\lambda+\Delta t}\right]\Delta\mathbf{u} \\ &\quad+\left[-\nabla\cdot\left(\mathbf{uu}-\frac{1}{\rho}\frac{2\lambda}{2\lambda+\Delta t}\tilde{\boldsymbol{\tau}}\right)^{n+\frac{1}{2}}-\frac{1}{\rho}\nabla p^{n-\frac{1}{2}}\right], \quad (4.129)\end{aligned}$$

which satisfies the proper formulations in the Newtonian (4.123) and elastic (4.124) limits. The rate-of-strain tensor from equation (4.127) now appears in the Laplacian term, and the discretization of this diffusion equation is provided below. The star in the superscript of (4.129) indicates that the preliminary velocity is not necessarily divergence-free. After $\mathbf{u}^{n+1,*}$ has been computed, a cell-centered projection described in Section 4.5.2 is applied immediately following the diffusion equation solver, and before the conservative source terms $\mathbf{S}_{U,\mathbf{M}}$ and $\mathbf{S}_{U,\mathbf{g}}$ are applied, to give the final cell-centered velocity.

Diffusion equation discretization: Each component of velocity in the diffusion equation takes the following form:

$$\frac{\partial\phi}{\partial t}=D\Delta\phi+f. \quad (4.130)$$

The solution to this diffusion equation is computed using a second-order, semi-implicit, L_0-stable method [78], which has been shown to be stable in the presence of fixed and moving boundaries in an EB geometry framework [43, 54, 71]. There is an additional advantage to using this method over the Crank-Nicolson scheme. The Crank-Nicolson discretization is not L_0-stable, that is, high frequency modes in the solution are not damped sufficiently, which can lead to numerical oscillations in the solution. The method used here is L_0-stable and avoids such oscillations.

Following [78], there are two steps to solving the diffusion equation, each in-

volving the solution of a Helmholtz operator. First, an intermediate solution ϕ' is computed:

$$\kappa_{\mathbf{i}}\left[\mathbf{I}-\eta_2 D\Delta_I^h\right]\phi' = \kappa_{\mathbf{i}}\left[\left(\mathbf{I}+\eta_3 D\Delta_I^h\right)\phi^n + \left(\mathbf{I}+\eta_4 D\Delta_H^h\right) f^{n+\frac{1}{2}}\Delta t\right], \tag{4.131}$$

followed by an equation which yields the time-advanced solution, ϕ^{n+1}:

$$\kappa_{\mathbf{i}}\left[\mathbf{I}-\eta_1 D\Delta_I^h\right]\phi^{n+1} = \kappa_{\mathbf{i}}\phi'. \tag{4.132}$$

Here, Δ_I^h and Δ_H^h are the discrete Laplacian operators with inhomogeneous and homogeneous boundary conditions, respectively. To solve these Helmholtz operators, first consider a general Helmholtz operator of the form:

$$(\mathbf{I}-D\Delta^h)\phi = b. \tag{4.133}$$

A linear equation is associated with each control volume, leading to a system of n linear equations, where n is the total number of control volumes in the domain. This linear system can be expressed as a matrix equation of the form $\boldsymbol{\mathcal{L}}\phi = b$. For regular cells away from domain boundaries, the linear equation associated with $V_{\mathbf{i}}$ uses the standard five-point Laplacian (4.79):

$$\phi_{\mathbf{i}} - D\left(\frac{\phi_{\mathbf{i}+\mathbf{e}_0}+\phi_{\mathbf{i}+\mathbf{e}_1}-4\phi_{\mathbf{i}}+\phi_{\mathbf{i}-\mathbf{e}_0}+\phi_{\mathbf{i}-\mathbf{e}_1}}{h^2}\right) = b_{\mathbf{i}}. \tag{4.134}$$

The same finite volume discretization for the Laplacian from (4.48) is used. As mentioned in Section 4.2, the Laplacian stencil contains a factor of the inverse volume fraction, $1/\kappa_{\mathbf{i}}$, so the Helmholtz discretizations are scaled by $\kappa_{\mathbf{i}}$ to cancel the potentially small denominator, providing numerical stability. To solve the resulting linear system, an iterative multigrid solver is used, in which the full linear system is solved by iteratively solving a series of smaller linear systems, as described in

Appendix Section A.4. Following [78], the parameters η_i are defined as:

$$\eta_1 = \frac{a - \sqrt{a^2 - 4a + 2}}{2}\Delta t, \tag{4.135}$$

$$\eta_2 = \frac{a + \sqrt{a^2 - 4a + 2}}{2}\Delta t, \tag{4.136}$$

$$\eta_3 = (1 - a)\Delta t, \tag{4.137}$$

$$\eta_4 = \left(\frac{1}{2} - a\right)\Delta t. \tag{4.138}$$

In order to minimize the truncation error while maintaining a second-order, L_0 stable method, $a = 2 - \sqrt{2} - \xi$, where $\xi = 1.0 \times 10^{-12}$ is on the order of machine precision. The same velocity boundary conditions are used as in the explicit Laplacian in equation (4.78). Note that the divergence terms in f are evaluated after the redistribution step, and despite the fact that the pressure gradient in f is time-lagged, the overall algorithm still shows second-order convergence.

Inverse deformation gradient source terms: After the cell-centered projection, the next step is to account for $\mathbf{S}_{U,\mathbf{g}}$. Recall from (4.33) that:

$$\mathbf{S}_{U,\mathbf{g}} = \left[\mathbf{u} \times \left(\nabla \times \mathbf{g}^T\right)\right]^T = \begin{bmatrix} u_1\left(\frac{\partial g_{01}}{\partial x_0} - \frac{\partial g_{00}}{\partial x_1}\right) \\ u_1\left(\frac{\partial g_{11}}{\partial x_0} - \frac{\partial g_{10}}{\partial x_1}\right) \\ -u_0\left(\frac{\partial g_{01}}{\partial x_0} - \frac{\partial g_{00}}{\partial x_1}\right) \\ -u_0\left(\frac{\partial g_{11}}{\partial x_0} - \frac{\partial g_{10}}{\partial x_1}\right) \end{bmatrix}. \tag{4.139}$$

The term $\tilde{\mathbf{g}}^{n+1}$ is defined as the inverse deformation gradient after the conservative update and redistribution step, and before accounting for the source terms. The

following discretization used in [56] for $\mathbf{S}_{U,\mathbf{g}}$ is:

$$\mathbf{g}_{\mathbf{i}}^{n+1} = \tilde{\mathbf{g}}_{\mathbf{i}}^{n+1} + \frac{\Delta t}{h} \begin{bmatrix} u_{1,\mathbf{i}}^{n+\frac{1}{2}} \left[\left(g_{01,\mathbf{i}+\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} - g_{01,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} \right) - \left(g_{00,\mathbf{i}+\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} - g_{00,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} \right) \right] \\ u_{1,\mathbf{i}}^{n+\frac{1}{2}} \left[\left(g_{11,\mathbf{i}+\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} - g_{11,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} \right) - \left(g_{10,\mathbf{i}+\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} - g_{10,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} \right) \right] \\ -u_{0,\mathbf{i}}^{n+\frac{1}{2}} \left[\left(g_{01,\mathbf{i}+\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} - g_{01,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} \right) - \left(g_{00,\mathbf{i}+\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} - g_{00,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} \right) \right] \\ -u_{0,\mathbf{i}}^{n+\frac{1}{2}} \left[\left(g_{11,\mathbf{i}+\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} - g_{11,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}^{n+\frac{1}{2}} \right) - \left(g_{10,\mathbf{i}+\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} - g_{10,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}^{n+\frac{1}{2}} \right) \right] \end{bmatrix}, \tag{4.140}$$

where in this equation, $\mathbf{u}_{\mathbf{i}}^{n+\frac{1}{2}}$ is the average of the initial and final velocity, $\mathbf{u}_{\mathbf{i}}^{n+\frac{1}{2}} = (\mathbf{u}_{\mathbf{i}}^{n} + \mathbf{u}_{\mathbf{i}}^{n+1})/2$. The face-centered, time-centered values of $\mathbf{g}$ are taken from the final MAC state.

Stress source terms: Recall from (4.33) that:

$$\mathbf{S}_{U,\mathbf{M}} = \mathbf{g} \cdot \left[\left(\frac{\mu_p}{\lambda} - a^2 \right) 2\mathbf{D} - \frac{1}{\lambda} \boldsymbol{\tau} \right] \cdot \mathbf{g}^T. \tag{4.141}$$

The term $\tilde{\mathbf{M}}^{n+1}$ is defined as value obtained after the redistribution step and before accounting for the source terms. The source terms for $\mathbf{M}$ are accounted for using a Crank-Nicolson discretization, as follows:

$$\begin{aligned} \mathbf{M}^{n+1} \quad = \quad & \tilde{\mathbf{M}}^{n+1} + \frac{\Delta t}{2} \left[\left(\mathbf{g} \cdot [(\frac{\mu_p}{\lambda} - a^2) 2\mathbf{D} - \frac{1}{\lambda} \boldsymbol{\tau}] \cdot \mathbf{g}^T \right)^n \right. \\ & \left. + \left(\mathbf{g} \cdot [(\frac{\mu_p}{\lambda} - a^2) 2\mathbf{D} - \frac{1}{\lambda} \boldsymbol{\tau}] \cdot \mathbf{g}^T \right)^{n+1} \right]. \end{aligned} \tag{4.142}$$

Since $\boldsymbol{\tau}^{n+1}$ appears on the right hand side of this equation, the terms can be algebraically rearranged so that the entire right hand side is a known quantity:

$$\mathbf{M}^{n+1} = \frac{2\lambda}{2\lambda+\Delta t}\tilde{\mathbf{M}}^{n+1} + \frac{\Delta t}{2\lambda+\Delta t}\Big[\big(\mathbf{g}\cdot[(\mu_p - a^2\lambda)2\mathbf{D} - \boldsymbol{\tau}]\cdot\mathbf{g}^T\big)^n + \big(\mathbf{g}\cdot[(\mu_p - a^2\lambda)2\mathbf{D} + a^2\mathbf{I}]\cdot\mathbf{g}^T\big)^{n+1}\Big]. \tag{4.143}$$

4.5 Projection Discretizations

This thesis uses the discretizations for the MAC and cell-centered projections used in [5] for solving the incompressible Navier-Stokes equations. The preliminary MAC velocities obtained from the transverse predictor Riemann problem (4.88), $\mathbf{u}^{n+\frac{1}{2},*}$, and the preliminary cell-centered velocities obtained from the solution to the diffusion equation (4.129), $\mathbf{u}^{n+1,*}$, do not necessarily satisfy the divergence constraint, $\nabla\cdot\mathbf{u} = 0$. A MAC projection is performed on $\mathbf{u}^{n+\frac{1}{2},*}$ and a cell-centered projection is performed on $\mathbf{u}^{n+1,*}$ to enforce this constraint. The projection operator, $\mathbf{P}$, operating on an arbitrary vector field $\mathbf{v}$, consists of the solution to the Poisson equation (3.5) followed by the update equation (3.6), a process which is abbreviated as:

$$\mathbf{P}(\mathbf{v}) = \mathbf{v} - \mathbf{Q}(\mathbf{v}), \tag{4.144}$$

$$\mathbf{Q}(\mathbf{v}) = \nabla\Delta^{-1}(\nabla\cdot\mathbf{v}). \tag{4.145}$$

A projection can be either exact or approximate. For an exact projection, the discrete divergence, gradient, and Laplacian are defined such that $\nabla\cdot(\nabla\phi) = \Delta\phi$, ensuring that $\mathbf{P}(\mathbf{v})$ is discretely divergence-free. Also, $\mathbf{P}(\nabla\phi) = 0$ and $\mathbf{P}(\mathbf{v}_\mathrm{d}) = \mathbf{v}_\mathrm{d}$, implying that $\mathbf{P}$ is idempotent, i.e., $\mathbf{P}^2 = \mathbf{P}$. Unfortunately, there are certain disadvantages for some exact projections. Consider a velocity field where both components of velocity are located at the cell-center, and define the divergence and

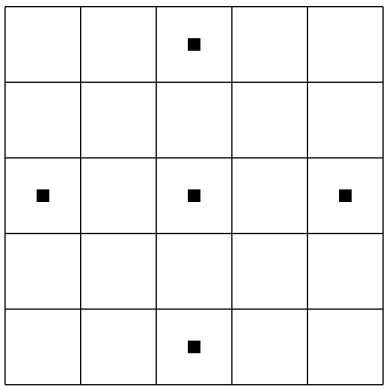

Figure 4.20: Stencil points for the spread out five-point Laplacian that arises in exact cell-centered projections.

gradient as the following centered difference approximations:

$$(\nabla \cdot \mathbf{u})_{\mathbf{i}} = \frac{u_{0,\mathbf{i}+\mathbf{e}_0} - u_{0,\mathbf{i}-\mathbf{e}_0}}{2h} + \frac{u_{1,\mathbf{i}+\mathbf{e}_1} - u_{1,\mathbf{i}-\mathbf{e}_1}}{2h}, \tag{4.146}$$

$$(\nabla \phi)_{d,\mathbf{i}} = \frac{\phi_{\mathbf{i}+\mathbf{e}_d} - \phi_{\mathbf{i}-e_d}}{2h}. \tag{4.147}$$

Using these definitions of the divergence and gradient, the stencil for the Laplacian, $\Delta\phi = \nabla \cdot (\nabla\phi)$, becomes a spread out version of the standard five-point Laplacian, as illustrated in Figure 4.20. This non-standard Laplacian operator creates four decoupled sets of stencil points which complicates the linear algebra used to solve the Poisson equation [46]. For an approximate projection, $\nabla \cdot (\nabla\phi) \neq \Delta\phi$, the resulting velocity field is not discretely divergence-free, and $\mathbf{P}^2 \neq \mathbf{P}$. Additionally, the divergence discretization can contain a null space which prevents certain oscillatory modes from being removed. However, approximate projections provide greater flexibility in the Laplacian discretization, often resulting in linear systems that are well conditioned for iterative solvers. For the case where both components of the velocity are located at cell-centers, it has been shown that an approximate projection using the centered difference discretizations for the divergence (4.146) and gradient (4.147), and a compact five-point stencil for the Laplacian can still lead to

a stable, second-order projection [46]. For the MAC projection, described in Section 4.5.1, only the normal components of velocity contribute to the divergence, which enables an exact projection discretization with a compact Laplacian stencil. The cell-centered projection, described in Section 4.5.2, is an approximate projection.

For the cases of inflow and outflow boundaries where $\mathbf{u}_\mathrm{d} \cdot \mathbf{n} \neq 0$, the boundary conditions for ϕ in the Poisson equation are modified using the ideas in [3]. At inflow, in order for the pressure gradient to respect the Dirichlet condition on the normal velocity, the condition $\mathbf{n} \cdot \nabla\phi = (\mathbf{u} - \mathbf{u}_\mathrm{d}) \cdot \mathbf{n}$ must be used. This is actually a generalized Dirichlet boundary condition that is valid even for no-flow conditions. At outflow, in order for the pressure gradient to respect the homogeneous Neumann condition on the normal velocity, the condition $\phi = 0$ is used. Physically, this imposes the condition that there are no net forces accelerating the fluid parallel to the outflow face.

4.5.1 MAC Projection

The first step in the MAC projection is to solve the Poisson equation (3.5). Similar to the Helmholtz operator in the diffusion equation solver, a linear equation is associated with each control volume leading to a matrix equation of the form $\boldsymbol{\mathcal{L}}\phi = b$ can be constructed. For regular cells away from domain boundaries, the linear equation associated with $V_\mathbf{i}$ uses the five-point Laplacian (4.79) and four-point divergence (4.4):

$$\frac{\phi_{\mathbf{i}+\mathbf{e}_0} + \phi_{\mathbf{i}+\mathbf{e}_1} - 4\phi_\mathbf{i} + \phi_{\mathbf{i}-\mathbf{e}_0} + \phi_{\mathbf{i}-\mathbf{e}_1}}{h^2} = \frac{u^{n+\frac{1}{2},*}_{0,\mathbf{i}+\frac{1}{2}\mathbf{e}_0} - u^{n+\frac{1}{2},*}_{0,\mathbf{i}-\frac{1}{2}\mathbf{e}_0}}{h} + \frac{u^{n+\frac{1}{2},*}_{1,\mathbf{i}+\frac{1}{2}\mathbf{e}_1} - u^{n+\frac{1}{2},*}_{1,\mathbf{i}-\frac{1}{2}\mathbf{e}_1}}{h}. \tag{4.148}$$

The linear equation associated with an arbitrary $V_\mathbf{i}$ is given as:

$$\kappa_\mathbf{i}(\Delta^h\phi)_\mathbf{i} = \kappa_\mathbf{i}\left(\nabla \cdot \mathbf{u}^{n+\frac{1}{2},*}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}\right)^{C,FC\to CC}_\mathbf{i}. \tag{4.149}$$

Similar to the diffusion equation solver, the Poisson equation discretization is scaled by $\kappa_{\mathbf{i}}$ to cancel the potentially small denominator. Again, the finite volume discretization for $\Delta^h \phi$ from (4.48) is used. To solve the resulting linear system, the same iterative multigrid solver used in the diffusion equation solver is used, as described in Appendix Section A.4.

The second step in the MAC projection is to update the velocity using equation (3.6) with the following discretization:

$$\mathbf{u}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{n+\frac{1}{2}} = \mathbf{u}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{n+\frac{1}{2},*} - (\nabla\phi)_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{CC\to FC}. \tag{4.150}$$

For regular cells away from domain boundaries, the normal gradient is the two-point difference approximation:

$$(\nabla\phi)_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{CC\to FC} = \frac{\phi_{\mathbf{i}+\mathbf{e}_d} - \phi_{\mathbf{i}}}{h}. \tag{4.151}$$

The transverse gradient is the following four-point difference approximation, which is the average of the two neighboring cell-centered transverse gradients:

$$(\nabla\phi)_{d',\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{CC\to FC} = \frac{1}{2}\left(\frac{\phi_{\mathbf{i}+\mathbf{e}_{d'}} - \phi_{\mathbf{i}-\mathbf{e}_{d'}}}{2h} + \frac{\phi_{\mathbf{i}+\mathbf{e}_d+\mathbf{e}_{d'}} - \phi_{\mathbf{i}+\mathbf{e}_d-\mathbf{e}_{d'}}}{2h}\right). \tag{4.152}$$

The general discretization for $(\nabla\phi)^{CC\to FC}$ is given in Appendix Section A.2.2.

The boundary conditions for velocity have already been enforced in the reconstruction algorithm in the transverse predictor Riemann problem. Note that when computing the divergence, only the normal velocity enters into the divergence stencil, and therefore specifying boundary conditions for the transverse velocity does not affect the divergence. As for the boundary conditions on ϕ, a homogeneous Neumann condition $\mathbf{n}\cdot\nabla\phi = 0$ is used at inflow and solid walls, and a homogeneous Dirichlet condition $\phi = 0$ is used at outflow. After the MAC projection, the normal velocity at outflow is overwritten to a value which forces the conservative divergence

in the cell to be zero.

4.5.2 Cell-Centered Projection and Pressure Update

Unlike the MAC projection, the cell-centered projection is an approximate projection and also provides an update for the pressure. Almgren, et al. [2] analyzed various cell-centered projection schemes, recommending the following scheme, which is also used in [77, 5].

$$\mathbf{v}^* = \mathbf{u}^{n+1,*} + \frac{\Delta t}{\rho}\nabla p^{n-\frac{1}{2}}, \tag{4.153}$$

$$\mathbf{u}^{n+1} = \mathbf{P}\left(\mathbf{v}^*\right), \tag{4.154}$$

$$\nabla p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t}\mathbf{Q}\left(\mathbf{v}^*\right). \tag{4.155}$$

Again, the Poisson equation in the projection operator results in n linear equations, which are solved with the same iterative multigrid solver used in the MAC projection. For regular cells away from domain boundaries, the linear equation associated with $V_{\mathbf{i}}$ uses the five-point Laplacian (4.79) and a new four-point divergence, which uses cell-centered stencil points instead of face-centered points:

$$\frac{\phi_{\mathbf{i}+\mathbf{e}_0} + \phi_{\mathbf{i}+\mathbf{e}_1} - 4\phi_{\mathbf{i}} + \phi_{\mathbf{i}-\mathbf{e}_0} + \phi_{\mathbf{i}-\mathbf{e}_1}}{h^2} = \frac{v^*_{0,\mathbf{i}+\mathbf{e}_0} - v^*_{0,\mathbf{i}-\mathbf{e}_0}}{2h} + \frac{v^*_{1,\mathbf{i}+\mathbf{e}_1} - v^*_{1,\mathbf{i}-\mathbf{e}_1}}{2h}. \tag{4.156}$$

The linear equation associated with an arbitrary $V_{\mathbf{i}}$ is given as:

$$\kappa_{\mathbf{i}}(\Delta^h\phi)_{\mathbf{i}} = \kappa_{\mathbf{i}}\left[\nabla\cdot\left(\mathbf{u}^{n+1,*} + \frac{\Delta t}{\rho}\nabla p^{n-\frac{1}{2}}\right)\right]^{C,CC\to CC}_{\mathbf{i}}, \tag{4.157}$$

$$\mathbf{u}^{n+1}_{\mathbf{i}} = \left(\mathbf{u}^{n+1,*}_{\mathbf{i}} + \frac{\Delta t}{\rho}\nabla p^{n-\frac{1}{2}}\right) - (\nabla\phi)^{CC\to CC}_{\mathbf{i}}, \tag{4.158}$$

$$\nabla p^{n+\frac{1}{2}}_{\mathbf{i}} = \frac{\rho}{\Delta t}(\nabla\phi)^{CC\to CC}_{\mathbf{i}}. \tag{4.159}$$

To compute the cell-centered conservative divergence of a cell-centered field, $(\nabla \cdot \mathbf{v})^{C,CC\to CC}$, the cell-centered values are first averaged to face-centers, and then a conservative divergence (4.44) is performed:

$$(\nabla \cdot \mathbf{v})_{\mathbf{i}}^{C,CC\to CC} = \left[\nabla \cdot \mathrm{Avg}(\mathbf{v})^{CC\to FC}\right]_{\mathbf{i}}^{C,FC\to CC}. \tag{4.160}$$

The operator $\mathrm{Avg}(\mathbf{v})^{CC\to FC}$ simply averages cell-centered values onto face-centers:

$$\mathrm{Avg}(\mathbf{v})_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}^{CC\to FC} = \frac{\mathbf{v}_{\mathbf{i}} + \mathbf{v}_{\mathbf{i}\pm\mathbf{e}_d}}{2}. \tag{4.161}$$

Boundary conditions need to be specified for the velocity and pressure gradient for the averaging operator used in $(\nabla \cdot \mathbf{v})^{C,CC\to CC}$. At inflow, the normal velocity is a prescribed Dirichlet value, and $\mathbf{n} \cdot \nabla p = 0$. At solid walls, the normal velocity is zero and $\mathbf{n} \cdot \nabla p = 0$. At outflow, the normal velocity and pressure gradient are extrapolated using a second-order linear extrapolation from the two nearest inward face-centers. At covered faces, $\mathrm{Avg}(\mathbf{v})^{CC\to FC}$ is not needed since covered face values do not enter into the conservative divergence stencil. Boundary conditions for ϕ are the same as for the MAC projection, that is, homogeneous Neumann $\mathbf{n}\cdot\nabla\phi$ at inflow and solid walls, and $\phi = 0$ at outflow. Once again, the discretization is scaled by $\kappa_{\mathbf{i}}$ to provide numerical stability in small cells.

As mentioned before, due to the approximate nature of this projection, certain oscillatory modes are not removed by this projection, requiring a filtering algorithm [46, 29]. Some examples of oscillatory modes, which contain divergence that is not detected by the cell-centered divergence operator, are illustrated in Figure 4.21. By choosing a divergence stencil other than the four-point stencil in the right-hand-side of (4.156), the oscillatory modes are suppressed by updating the velocity as follows:

$$\mathbf{u} = \mathbf{u} + \zeta\nabla(\nabla \cdot \mathbf{u}), \tag{4.162}$$

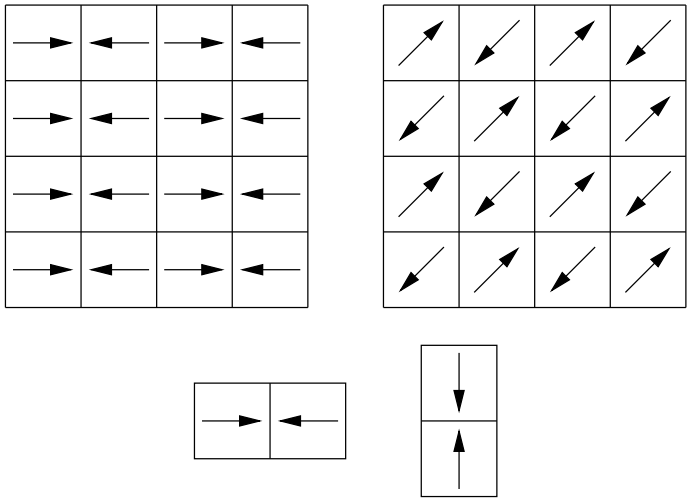

Figure 4.21: (Above) Sample oscillatory velocity modes that are not detected by the divergence operator in the cell-centered projection. (Below) Two-point velocity modes in the x_0 and x_1 directions that form a basis for all oscillatory modes not detected by the cell-centered projection.

where ζ is a constant damping factor. For regular cells away from domain boundaries, the choice has been made to use the "edge-centered divergence" stencil [68], illustrated in Figure 4.22:

$$(\nabla \cdot \mathbf{u})_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{u_{d,\mathbf{i}+\mathbf{e}_d} - u_{d,\mathbf{i}}}{h} + \frac{1}{2}\left(\frac{u_{d',\mathbf{i}+\mathbf{e}_{d'}} - u_{d',\mathbf{i}-\mathbf{e}_{d'}}}{2h} + \frac{u_{d',\mathbf{i}+\mathbf{e}_d+\mathbf{e}_{d'}} - u_{d',\mathbf{i}+\mathbf{e}_d-\mathbf{e}_{d'}}}{2h}\right), \tag{4.163}$$

$$\left[\nabla(\nabla \cdot \mathbf{u})_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}\right]_{d,\mathbf{i}} = \frac{(\nabla \cdot \mathbf{u})_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - (\nabla \cdot \mathbf{u})_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}{h}. \tag{4.164}$$

For this discretization of (4.162), the algorithm uses $\zeta = h^2/5$ in order to guarantee stability and guarantee the damping of the oscillatory modes [29]. The general discretization for the edge-centered divergence stencil in the presence of domain and embedded boundaries is given in Appendix Section A.5.

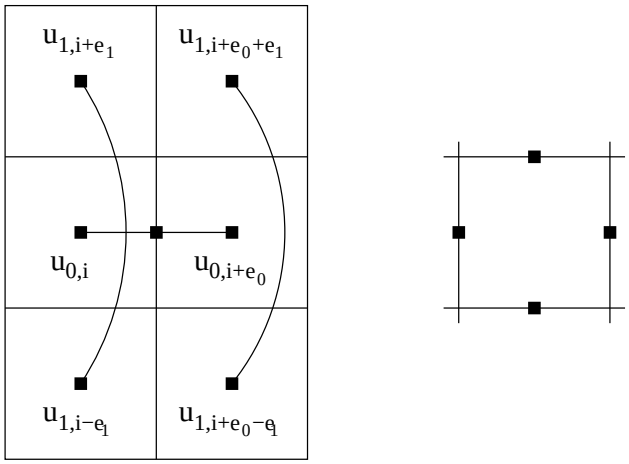

Figure 4.22: (Left) Stencil points for the edge-centered divergence for regular cells. (Right) Stencil points for the cell-centered gradient of the edge-centered divergence.

4.6 Time Step and Initialization

Since the source terms are handled implicitly, they do not provide any stability constraint on the time step. Therefore, the time step is limited by the CFL stability constraint for the Godunov method [24]:

$$\Delta t \leq \mathrm{CFL}\frac{h}{|\Lambda_{\max}|}; \qquad 0 < \mathrm{CFL} < 1, \tag{4.165}$$

where $\Lambda_{\max}$ is the eigenvalue of largest magnitude in the unpartitioned primitive coefficient matrices over the entire domain, including the boundary conditions. This gives a CFL time step condition that can be compared directly to [77].

At the beginning of each time step, $\mathbf{g}$ is reinitialized to $\mathbf{I}$, since the complete evolution of $\mathbf{g}$ is not required to advance the solution for $\mathbf{u}$ and $\boldsymbol{\tau}$. As mentioned in Section 4.1.1, a^2 is also reinitialized at the beginning of each time step. Therefore, after $\mathbf{g}$ and a^2 have been reinitialized, $\mathbf{M}$ must be reinitialized as well.

At the beginning of each simulation, the pressure gradient is initialized using a priming procedure. After choosing an initial velocity field, a cell-centered projection

and filter is performed to ensure the velocity is divergence-free. Next, the initial stress is set to zero, $\mathbf{g}$ is set to $\mathbf{I}$, and the pressure gradient is initialized to a problem specific value. Next, three pressure priming iterations are performed, in which the solution for a full time step is computed, yet only the pressure gradient is updated and the total time is not advanced.

5 Results

Results are now presented for two groups of tests which demonstrate the robustness and accuracy of the method. The first group of tests is the confined vortex flow of three types of Oldroyd-B fluids, in which the velocity is initialized to a smooth vortex function inside both a rectangular and circular geometry. The second group of tests is the flow of two real experimental fluids [39] in both an abrupt contraction and past a sphere.

For each test except for the abrupt contraction, a solution error test is performed. The abrupt contraction test is used for comparison to experimental data, and the solution error properties are adversely affected by the sharp geometry. In the solution error test, the solution is first computed on a coarse domain to a specified time, t, using a fixed time step. Next, both Δt and h are refined by successive factors of 2 to compute the solution at time t at a medium and fine resolution. To get the error between the medium and fine solutions, $\mathcal{E}_{\text{med-fine}}$, the fine solution is coarsened using a volume weighted average and the solutions are subtracted directly. This process is repeated to obtain the error between the coarse and medium solutions, $\mathcal{E}_{\text{coar-med}}$. The L^∞ norm is defined as the maximum value of $|\mathcal{E}_\mathbf{i}|$ on the domain. The L^1 and L^2 norms are defined as:

$$L^1(\mathcal{E}) = \frac{1}{|\Omega|}\int_\Omega |\mathcal{E}| dV = \frac{1}{\sum_\Omega \kappa_\mathbf{i}} \sum_\Omega \kappa_\mathbf{i} |\mathcal{E}_\mathbf{i}|, \tag{5.1}$$

$$L^2(\mathcal{E}) = \left(\frac{1}{|\Omega|}\int_\Omega \mathcal{E}^2 dV\right)^{\frac{1}{2}} = \left(\frac{1}{\sum_\Omega \kappa_\mathbf{i}} \sum_\Omega \kappa_\mathbf{i} \mathcal{E}_\mathbf{i}^2\right)^{\frac{1}{2}}. \tag{5.2}$$

The order of convergence corresponding to the L^N norm, P^N, is given by:

$$P^N = \frac{\log\left(\frac{L^N(\mathcal{E}_{\text{coar-med}})}{L^N(\mathcal{E}_{\text{med-fine}})}\right)}{\log 2}. \tag{5.3}$$

5.1 Confined Vortex Flow

For the confined vortex test problems, the first fluid is a Maxwell (highly elastic) fluid, characterized by having no solvent viscosity, a non-zero polymeric viscosity, and a non-zero relaxation time. The second fluid is a Newtonian fluid, characterized by having a non-zero solvent viscosity, no polymeric viscosity, and relaxation time of zero. The third fluid is a "hybrid" fluid [77], which is a Maxwell fluid with an added solvent viscosity.

For the rectangular geometry, the computational domain has $l = w = 2.0$. The rectangular box has dimensions $l = 1.7$, and $w = 1.0$, and has been rotated 45° to maximize the amount of fluid in the computational domain. The initial vortex velocity profile has been chosen to be sufficiently smooth at the vortex edge, given by the function $u_\theta(r) = 2.56[(r/0.45)(1 - r/0.45)]^4 H(0.45 - r)$, where r is the distance to the center of the box and H is the Heaviside step function. This gives a maximum initial speed of $|\mathbf{u}| = 1.0$ at $r = 0.225$ (see Figure 5.1, top-left). For all the images in this thesis corresponding to the angled box geometry, the output has been rotated so the variables are seen with respect to the normal (length-wise) and transverse (width-wise) directions, which is done using a rotation similar to that in Section 4.3.4. The initial pressure is set to zero. The characteristic speed, U, is defined as the maximum initial velocity and the characteristic length, L, is defined as the width of the box.

For the circular geometry, the computational domain has $l = w = 1.0$ and the circle has radius $r = 0.45$ to maximize the amount of fluid in the computational domain. The initial velocity profile is $u_\theta(r) = 2.56[(r/0.4)(1 - r/0.4)]^4 H(0.4 - r)$, which gives a maximum initial speed of $|\mathbf{u}| = 1.0$ at $r = 0.2$ (see Figure 5.3, top-left). The initial pressure is set to zero. The characteristic speed, U, is defined as the maximum initial velocity and the characteristic length, L, is defined as the diameter of the circle.

5.1.1 Maxwell Fluid

For the Maxwell fluid, the rheological parameters are $\mu_s = 0, \mu_p = 1.0, \lambda = 1.0$, and $\rho = 1.0$. This gives the dimensionless parameters $\mathrm{Re} = 1.0, \mathrm{We} = 1.0$, and $\mathrm{Ma} = 1.0$ for the rectangular box geometry. The coarse domain resolution is 128×128 cells and the coarse time step is 3.2×10^{-3}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 5.1. The elastic wave propagation and reflection off the walls is clearly visible. The transverse velocity, stress, and pressure corresponding to the final image of normal velocity are shown in Figure 5.2. The solution error convergence after 400 fine time steps is given in Table 5.1.

For Maxwell fluids, it has been observed here that additional cell-centered filtering steps must be used to prevent the buildup of divergent modes near cells with small volume fractions. In the other flow regimes, the non-zero solvent viscosity present in the diffusion equation solver smooths the velocity and helps eliminate the divergent modes so that additional filtering steps are not required. The approach taken here to stabilize the method is to perform 1 filter iteration per time step at the coarse resolution, 2 iterations at the medium resolution, and 4 iterations at the finest resolution. The additional filters are not required for the other flow regimes, yet are still performed so the results are consistent with the case of the Maxwell fluid.

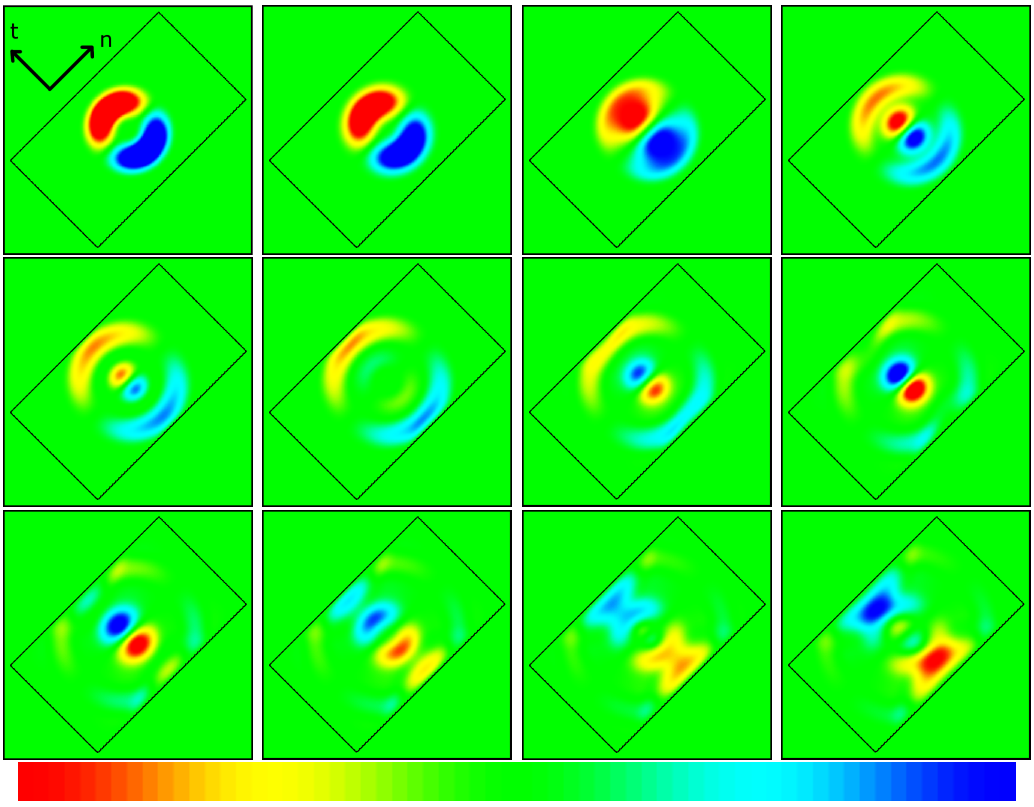

Figure 5.1: Time-dependent normal velocity u_n profiles of a Maxwell fluid with a vortex initial condition in an enclosed box. The grid resolution is 256×256 with 24 time step increments using $\Delta t = 1.6 \times 10^{-3}$. Scale: -0.5 (red) to 0.5 (blue).

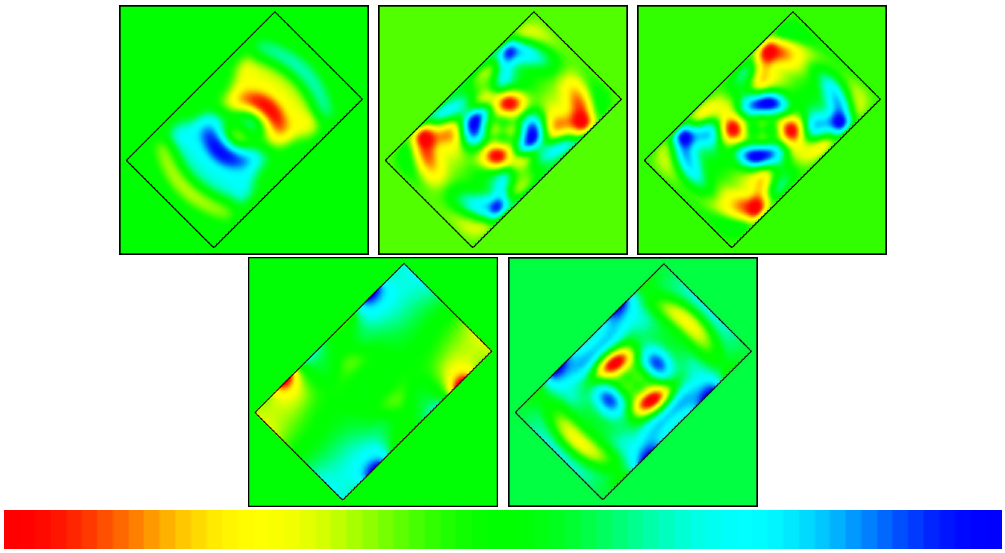

Figure 5.2: Profiles for a Maxwell fluid in an enclosed box at $t = 0.4288$ (corresponding to the last image in Figure 5.1.) In order, clockwise from top-left: transverse velocity u_t, -0.5 (red) to 0.5 (blue); normal stress τ_{nn}, -0.21 (red) to 0.31 (blue); normal stress τ_{tt}, -0.21 (red) to 0.28 (blue); shear stress τ_{tn}, -0.46 (red) to 0.33 (blue); hydrostatic pressure p, 0 (red) to 0.656 (blue).

N	Variable	Coarse Error	Fine Error	Order
1	u_0	9.90e-04	2.69e-04	1.88
1	u_1	9.62e-04	2.63e-04	1.87
1	τ_{00}	1.24e-03	3.06e-04	2.02
1	τ_{10}	1.38e-03	3.40e-04	2.02
1	τ_{11}	1.37e-03	3.39e-04	2.01
1	p	1.04e-03	2.68e-04	1.96
2	u_0	1.65e-03	4.34e-04	1.93
2	u_1	1.66e-03	4.23e-04	1.97
2	τ_{00}	1.89e-03	4.78e-04	1.98
2	τ_{10}	3.06e-03	6.93e-04	2.14
2	τ_{11}	3.27e-03	8.36e-04	1.97
2	p	2.52e-03	4.78e-04	2.40
∞	u_0	4.08e-02	6.32e-03	2.69
∞	u_1	4.15e-02	6.98e-03	2.57
∞	τ_{00}	5.11e-02	1.09e-02	2.23
∞	τ_{10}	8.36e-02	2.77e-02	1.59
∞	τ_{11}	1.45e-01	3.73e-02	1.95
∞	p	7.76e-02	1.15e-02	2.75

Table 5.1: Solution error convergence rates for a Maxwell fluid with a vortex initial condition inside a rectangular geometry.

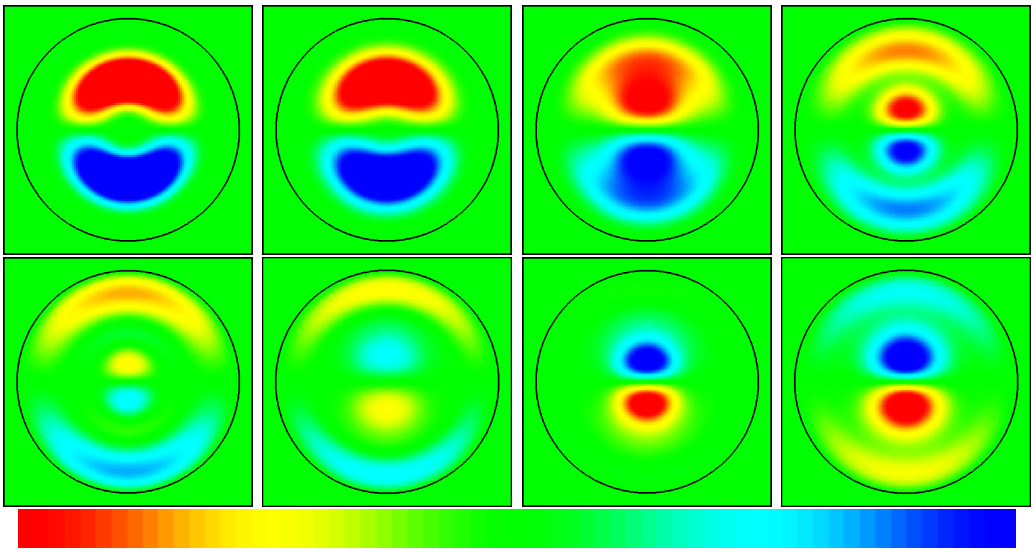

Figure 5.3: Time-dependent u_0 profiles of a Maxwell fluid with a vortex initial condition in an enclosed circle. The grid resolution is 128×128 with 24 time step increments using $\Delta t = 1.6 \times 10^{-3}$. Scale: -0.50 (red) to 0.50 (blue).

The same rheological parameters are used for the circular geometry, leading to dimensionless parameters $\mathrm{Re} = 0.9, \mathrm{We} = 0.9$, and $\mathrm{Ma} = 1.0$. The time-dependent u_0 profiles are shown in Figure 5.3. Again, the elastic wave propagation and reflection off the walls is easily visible. The u_1 component of velocity, stress, and pressure corresponding to the final image of u_0 are shown in Figure 5.4. The solution error convergence after 400 fine time steps is given in Table 5.2.

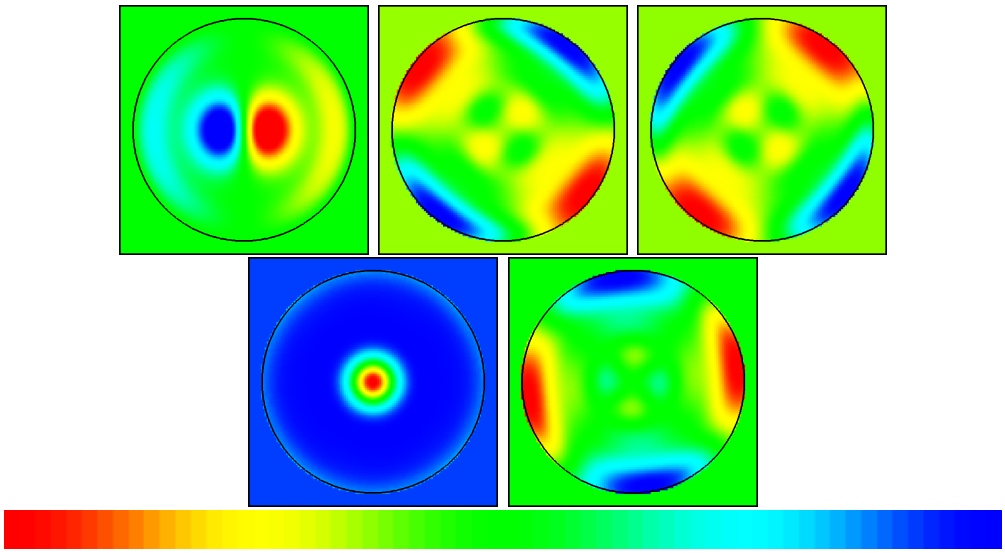

Figure 5.4: Profiles for a Maxwell fluid in an enclosed circle at $t = 4.288 \times 10^{-3}$ (corresponding to the last image in Figure 5.3.) In order, clockwise from top-left: u_1, -0.50 (red) to 0.50 (blue); normal stress τ_{00}, -0.38 (red) to 0.67 (blue); normal stress τ_{11}, -0.38 (red) to 0.67 (blue); shear stress τ_{10}, -0.53 (red) to 0.53 (blue); hydrostatic pressure p, 0 (red) to 0.55 (blue).

N	Variable	Coarse Error	Fine Error	Order
1	u_0	2.00e-03	5.70e-04	1.81
1	u_1	2.05e-03	6.14e-04	1.74
1	τ_{00}	2.01e-03	6.87e-04	1.55
1	τ_{10}	1.62e-03	6.87e-04	1.39
1	τ_{11}	2.03e-03	6.88e-04	1.56
1	p	1.49e-03	5.62e-04	1.40
2	u_0	3.06e-03	1.01e-03	1.59
2	u_1	3.15e-03	1.08e-03	1.55
2	τ_{00}	3.09e-03	1.02e-03	1.60
2	τ_{10}	2.33e-03	8.78e-04	1.41
2	τ_{11}	3.00e-03	1.00e-03	1.58
2	p	2.19e-03	8.60e-04	1.35
∞	u_0	3.11e-02	1.66e-02	0.91
∞	u_1	3.31e-02	1.64e-02	1.01
∞	τ_{00}	4.07e-02	2.24e-02	0.86
∞	τ_{10}	4.15e-02	1.94e-02	1.09
∞	τ_{11}	3.68e-02	2.31e-02	0.67
∞	p	3.04e-02	2.16e-02	0.49

Table 5.2: Solution error convergence rates for a Maxwell fluid with a vortex initial condition inside a circular geometry.

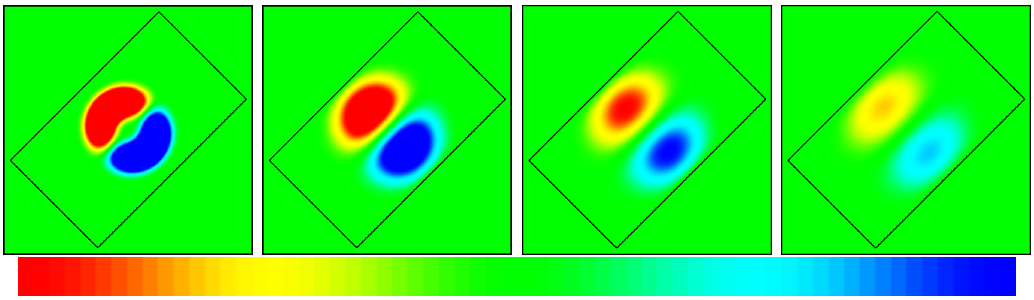

Figure 5.5: Time-dependent normal velocity u_n profiles of a Newtonian fluid with a vortex initial condition in an enclosed box. The grid resolution is 256×256 with 30 time step increments using $\Delta t = 3.0 \times 10^{-4}$. Scale: -0.25 (red) to 0.25 (blue).

5.1.2 Newtonian Fluid

For the Newtonian fluid, the rheological parameters are $\mu_s = 1.0, \mu_p = 0.0, \lambda = 1.0 \times 10^{-11}$, and $\rho = 1.0$ leading to dimensionless parameters Re = 1.0 and We = 0.0 for the rectangular box geometry. Since $\mu_p = 0$, the polymeric stress remains zero at all times. The coarse domain resolution is 64×64 and the coarse time step is 1.5×10^{-2}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 5.5, in which the vortex spreads out to fill the box and decays over time. The transverse velocity and pressure corresponding to the final image of normal velocity are shown in Figure 5.6. The solution error convergence after 20 fine time steps is given in Table 5.3. The reason why so few time steps are chosen is because the fluid velocity has already decayed to less than two percent of the initial value.

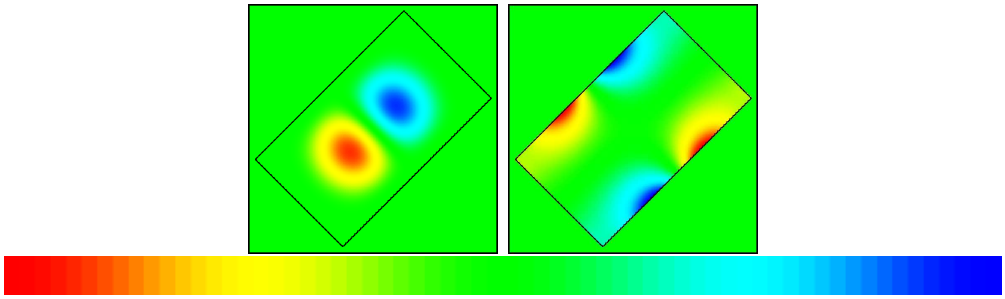

Figure 5.6: Profiles for a Newtonian fluid in an enclosed box at $t = 2.7 \times 10^{-2}$ (corresponding to the last image in Figure 5.5.) (Left) Transverse velocity u_t, -0.15 (red) to 0.15 (blue); (Right) hydrostatic pressure p, 0 (red) to 1.52 (blue).

N	Variable	Coarse Error	Fine Error	Order
1	u_0	8.44e-04	1.68e-04	2.33
1	u_1	8.39e-04	1.68e-04	2.32
1	p	1.69e-02	3.15e-03	2.42
2	u_0	1.14e-03	2.26e-04	2.33
2	u_1	1.13e-03	2.26e-04	2.33
2	p	3.07e-02	6.63e-03	2.21
∞	u_0	9.61e-03	2.35e-03	2.03
∞	u_1	9.66e-03	2.35e-03	2.04
∞	p	1.31e-01	4.13e-02	1.67

Table 5.3: Solution error convergence rates for a Newtonian fluid with a vortex initial condition inside a rectangular geometry.

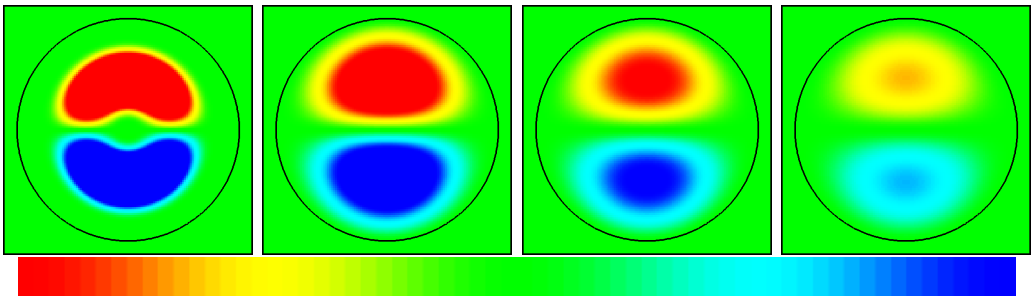

Figure 5.7: Time-dependent u_0 velocity u_0 profiles of a Newtonian fluid with a vortex initial condition in an enclosed circle. 20 time step increments, $\Delta t = 3.0 \times 10^{-4}$, -0.25 (red) to 0.25 (blue).

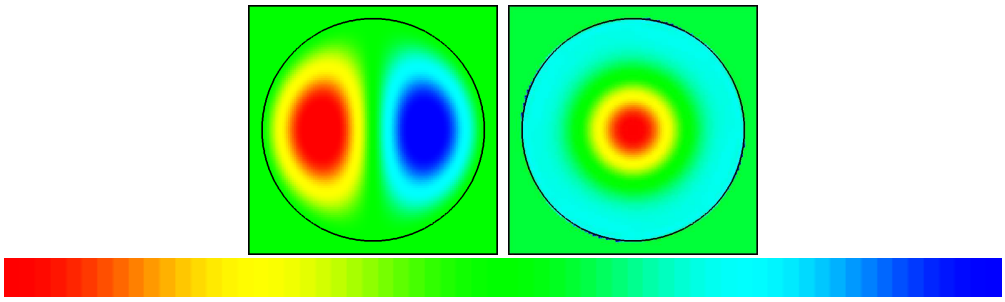

Figure 5.8: Profiles for a Newtonian fluid in an enclosed circle at $t = 1.8 \times 10^{-2}$ (corresponding to the last image in Figure 5.7.) (Left) u_1, -0.15 (red) to 0.15 (blue); (Right) hydrostatic pressure p, 0 (red) to 0.52 (blue).

The same rheological parameters are used for the circular geometry, leading to dimensionless parameters Re = 0.9 and We = 0. The time-dependent normal velocity is shown in Figure 5.7. Similar to the rectangular box case, the vortex spreads out to fill the circle and decays over time. The transverse velocity and pressure corresponding to the final image of normal velocity are shown in Figure 5.8. The solution error convergence after 20 fine time steps is given in Table 5.4.

N	Variable	Coarse Error	Fine Error	Order
1	u_0	4.06e-04	9.44e-05	2.10
1	u_1	4.06e-04	9.44e-05	2.10
1	p	5.12e-03	1.58e-04	5.02
2	u_0	4.88e-04	1.16e-04	2.07
2	u_1	4.88e-04	1.16e-04	2.07
2	p	1.04e-02	2.48e-04	5.40
∞	u_0	1.06e-03	5.51e-04	0.95
∞	u_1	1.06e-03	5.51e-04	0.95
∞	p	7.77e-02	1.13e-03	6.11

Table 5.4: Solution error convergence rates for a Newtonian fluid with a vortex initial condition inside a circular geometry.

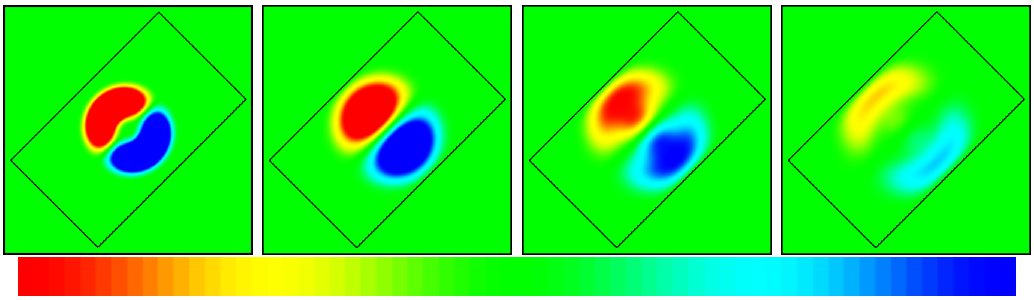

Figure 5.9: Time-dependent normal velocity u_n profiles of a hybrid fluid with a vortex initial condition in an enclosed box. The grid resolution is 256×256 with 30 time step increments using $\Delta t = 1.6 \times 10^{-3}$. Scale: -0.25 (red) to 0.25 (blue).

5.1.3 Hybrid Fluid

For the hybrid fluid, the rheological parameters are $\mu_s = 0.1, \mu_p = 0.9, \lambda = 1.0$, and $\rho = 1.0$ leading to dimensionless parameters $\mathrm{Re} = 1.0, \mathrm{We} = 1.0$, and $\mathrm{Ma} = 1.05$ for the rectangular box geometry. The initial stress is set to zero. The coarse domain resolution is 128×128 and the coarse time step is 6.4×10^{-3}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 5.9. As is the case with the Newtonian fluid, the vortex spreads out and decays over time, with a different shape than the Newtonian case. The transverse velocity, stress, and pressure corresponding to the final image of normal velocity are shown in Figure 5.10. The solution error convergence after 400 fine time steps is given in Table 5.5.

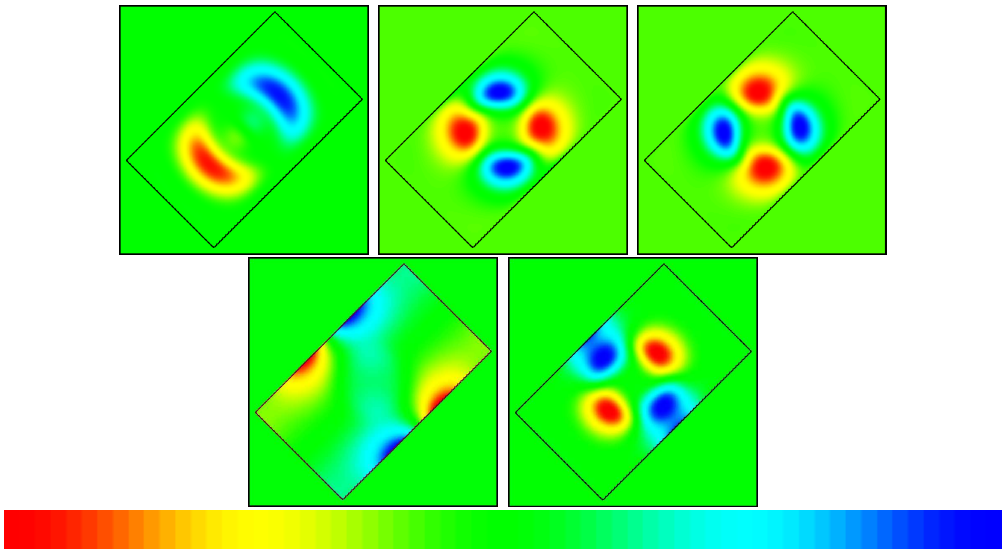

Figure 5.10: Profiles for a hybrid fluid in an enclosed box at $t = 0.144$ (corresponding to the last image in Figure 5.9.) In order, clockwise from top-left: transverse velocity u_t, -0.15 (red) to 0.15 (blue); normal stress τ_{nn}, -0.25 (red) to 0.37 (blue); normal stress τ_{tt}, -0.25 (red) to 0.37 (blue); shear stress τ_{tn}, -0.30 (red) to 0.29 (blue); hydrostatic pressure p, 0 (red) to 0.55 (blue).

N	Variable	Coarse Error	Fine Error	Order
1	u_0	3.66e-04	8.52e-05	2.10
1	u_1	3.80e-04	8.83e-05	2.11
1	τ_{00}	4.86e-04	1.29e-04	1.91
1	τ_{10}	6.09e-04	1.73e-04	1.82
1	τ_{11}	4.87e-04	1.31e-04	1.89
1	p	7.25e-04	2.50e-04	1.53
2	u_0	7.58e-04	1.96e-04	1.95
2	u_1	7.74e-04	2.01e-04	1.94
2	τ_{00}	7.97e-04	2.44e-04	1.71
2	τ_{10}	1.04e-03	3.00e-04	1.79
2	τ_{11}	8.41e-04	2.88e-04	1.54
2	p	1.08e-03	3.87e-04	1.48
∞	u_0	1.29e-02	6.78e-03	0.92
∞	u_1	1.30e-02	6.87e-03	0.93
∞	τ_{00}	5.68e-03	2.66e-03	1.09
∞	τ_{10}	9.44e-03	8.63e-03	0.13
∞	τ_{11}	6.41e-03	4.71e-03	0.44
∞	p	4.89e-03	6.67e-03	-0.45

Table 5.5: Solution error convergence rates for a hybrid fluid with a vortex initial condition inside a rectangular geometry.

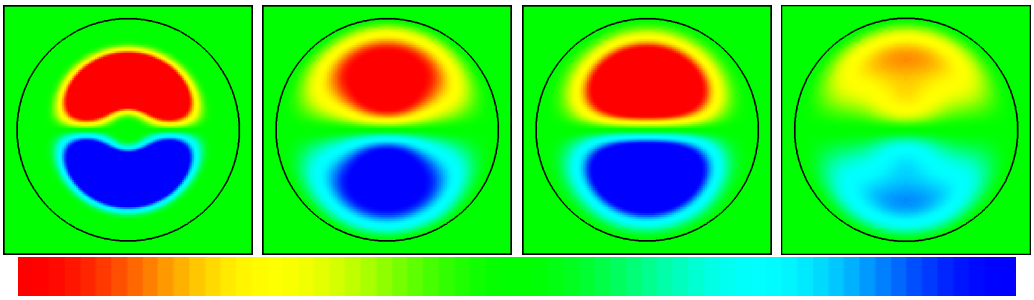

Figure 5.11: Time-dependent u_0 profiles of a hybrid fluid with a vortex initial condition in an enclosed circle. The grid resolution is 128×128 with 20 time step increments, $\Delta t = 1.6 \times 10^{-3}$, -0.25 (red) to 0.25 (blue).

The same rheological parameters are used for the circular geometry, leading to dimensionless parameters $\mathrm{Re} = 0.9, \mathrm{We} = 0.9$, and $\mathrm{Ma} = 1.05$. The time-dependent u_0 component of velocity is shown in Figure 5.11. Similar to the rectangular box case, the vortex spreads out to fill the circle and decays over time. The transverse velocity, stress, and pressure corresponding to the final image of u_0 are shown in Figure 5.12. The solution error convergence after 400 fine time steps is given in Table 5.6.

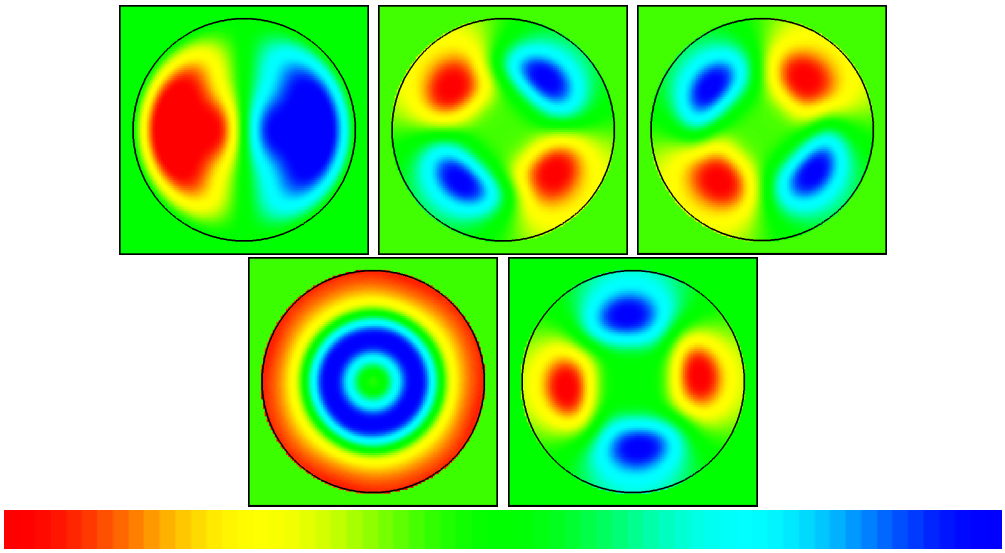

Figure 5.12: Profiles for a hybrid fluid in an enclosed circle at $t = 0.096$ (corresponding to the last image in Figure 5.11). In order, clockwise from top-left: u_1, -0.15 (red) to 0.15 (blue); normal stress τ_{00}, -0.25 (red) to 0.35 (blue); normal stress τ_{11}, -0.25 (red) to 0.35 (blue); shear stress τ_{10}, -0.30 (red) to 0.30 (blue); hydrostatic pressure p, 0 (red) to 0.040 (blue).

N	Variable	Coarse Error	Fine Error	Order
1	u_0	1.21e-04	3.05e-05	1.99
1	u_1	1.27e-04	3.40e-05	1.91
1	τ_{00}	2.66e-04	6.62e-05	2.01
1	τ_{10}	3.95e-04	1.25e-04	1.66
1	τ_{11}	2.54e-04	6.39e-05	1.99
1	p	3.94e-04	1.40e-04	1.50
2	u_0	2.23e-04	6.86e-05	1.70
2	u_1	2.27e-04	7.12e-05	1.67
2	τ_{00}	3.53e-04	9.91e-05	1.83
2	τ_{10}	4.94e-04	1.62e-04	1.60
2	τ_{11}	3.44e-04	9.73e-05	1.82
2	p	4.82e-04	1.72e-04	1.49
∞	u_0	2.31e-03	1.15e-03	1.00
∞	u_1	2.30e-03	1.15e-03	1.00
∞	τ_{00}	4.47e-03	2.13e-03	1.07
∞	τ_{10}	3.78e-03	1.64e-03	1.21
∞	τ_{11}	4.75e-03	2.28e-03	1.06
∞	p	2.19e-03	1.90e-03	0.20

Table 5.6: Solution error convergence rates for a hybrid fluid with a vortex initial condition inside a circular geometry.

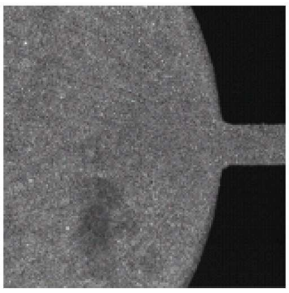

Figure 5.13: Image of the contraction channel geometry used in microfluidic experiments [39]. The downstream contraction has width $w = 0.013$ [cm].

5.2 Experimental Fluids

The second group of tests involves two real fluids from microfluidic experiments designed to examine the differences between Newtonian and viscoelastic fluid behavior in contraction geometries [39]. The first fluid is distilled water and the second fluid is a semi-dilute (400 [ug/mL]) concentration of DNA molecules from the species bacteriophage lambda (λ-DNA) added to a 1X TAE Newtonian solvent [39]. This particular DNA was chosen for experimentation because its structure is well-known and rheological properties have been widely studied [62, 72]. In the experiments, the fluids were pumped from a large reservoir into a downstream contraction, as shown in Figure 5.13. In this numerical simulation, the experimental device will be approximated as a 2:1 abrupt contraction to obtain a qualitative comparison between the downstream flow profiles. Additionally, the numerical convergence rates for the flow past a sphere for both of these fluids will be computed. In both of these computational tests, the fluid rheology, length scales, and velocity scales will match the experimental setup.

5.2.1 Experimental Data

In the experiments, fluid flows from a large reservoir into a downstream contraction of width $w = 0.013$ [cm] with an average downstream velocity of 0.042 [cm/s]

for the distilled water and 0.085 [cm/s] for the DNA fluid. The experimentally measured rheological parameters for the distilled water are $\mu_s = 0.01$ [cP], $\mu_p = 0.0$ [cP], $\lambda = 1.0 \times 10^{-11}$ [s], and $\rho = 1.0$ [g/cm^3]. The experimentally measured rheological parameters for the DNA fluid are $\mu_s = 0.01$ [cP], $\mu_p = 0.05919$ [cP], $\lambda = 0.91$ [s], and $\rho = 1.0$ [g/cm^3]. In the numerical simulations, both fluid velocities will match the distilled water experiment. The characteristic velocity, U, is the average velocity in the downstream contraction and the characteristic length, L, is half the width of the downstream contraction leading to computational dimensionless parameters of $\mathrm{Re} = 2.7 \times 10^{-2}$ and $\mathrm{We} = 0$ for the distilled water and $\mathrm{Re} = 4.0 \times 10^{-3}, \mathrm{We} = 5.9$, and $\mathrm{Ma} = 0.16$ for the DNA fluid.

The experimentally obtained transverse velocity profiles for this configuration is illustrated in Figure 5.14. The top image indicates the location where the transverse velocities were measured. The entrance length is clearly longer for the DNA fluid, as can be seen by the non-zero transverse velocity profile at point "D", whereas the transverse velocity profile for distilled water has a much smaller magnitude upstream at point "C".

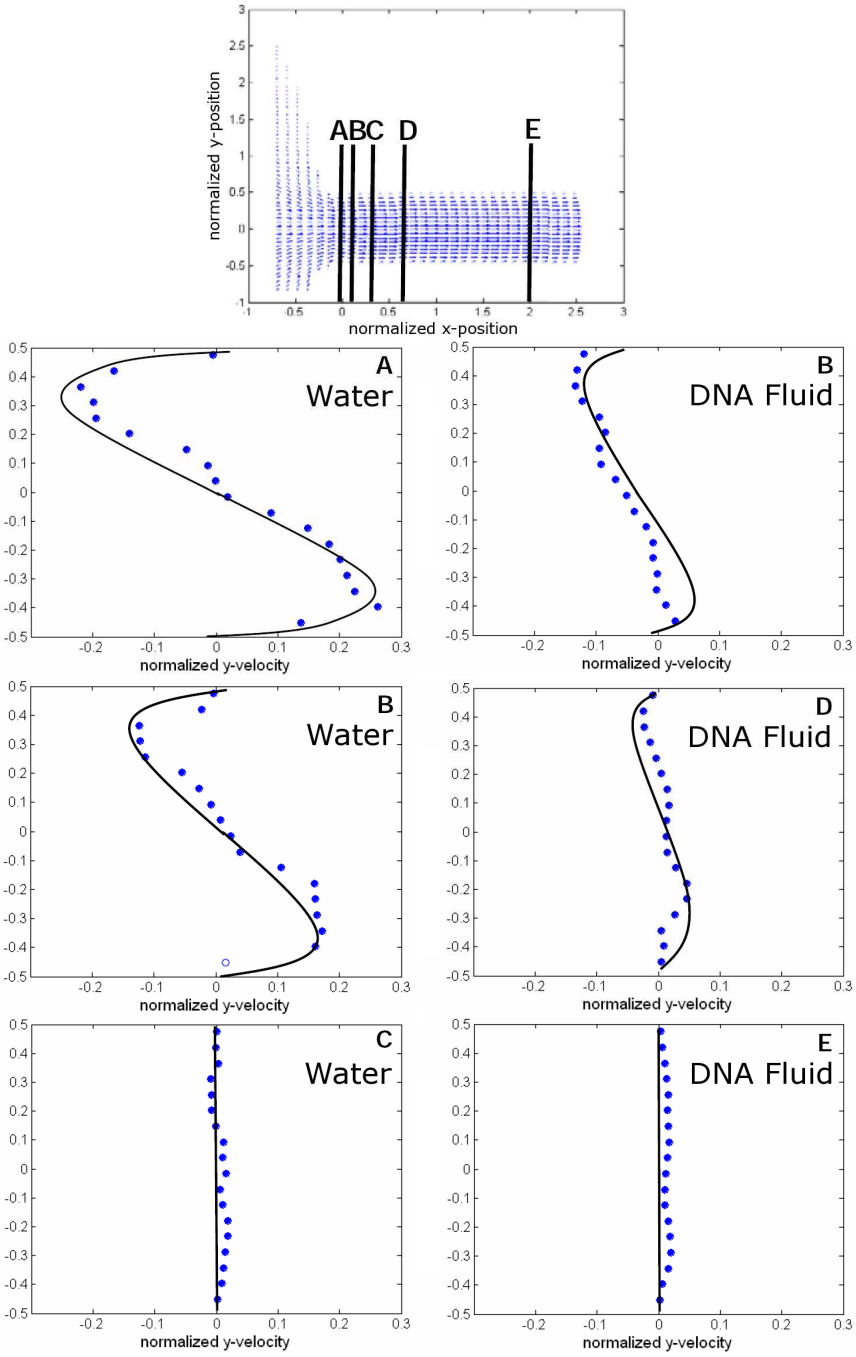

Figure 5.14: Experimentally obtained transverse velocity profiles [39] for distilled water and the DNA fluid at various locations downstream of the contraction in Figure 5.13, as indicated in the top image. The solid lines represent best-fit curves for the blue data points.

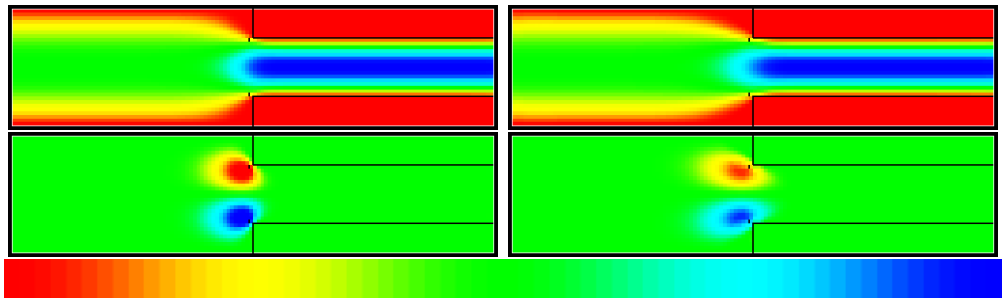

Figure 5.15: Computed normal and transverse velocity profiles of distilled water (Left) and DNA fluid (Right) in a 2:1 abrupt contraction geometry at $t = 0.15$ [s], u_0, 0 (red) to 0.063 (blue) [cm/s]; u_1, -0.01 (red) to 0.01 (blue) [cm/s].

5.2.2 Abrupt Contraction Geometry

In these numerical simulations, units are included since the geometrical and rheological parameters are obtained from the experiments. The computational domain has $l = 0.104$ [cm] and $w = 0.026$ [cm], and a 2:1 abrupt contraction is placed at the center of the domain, providing a downstream contraction width of $w = 0.013$ [cm]. The initial velocity, stress, and pressure as well as the inflow velocity and stress are set to the analytic solution for fully developed Poiseuille flow in the absence of the contraction. The grid resolution is 128×32 cells. The solution is computed to $t = 0.15$ [s] in order to allow the solution to fully develop.

Figure 5.15 illustrates the computed normal and transverse velocity profiles of both distilled water and the DNA fluid. The entrance lengths are clearly greater for the DNA fluid, as is examined in more detail in Figure 5.16, which illustrates the computed transverse velocity profiles at various downstream locations. This is in good qualitative agreement with the experimental results.

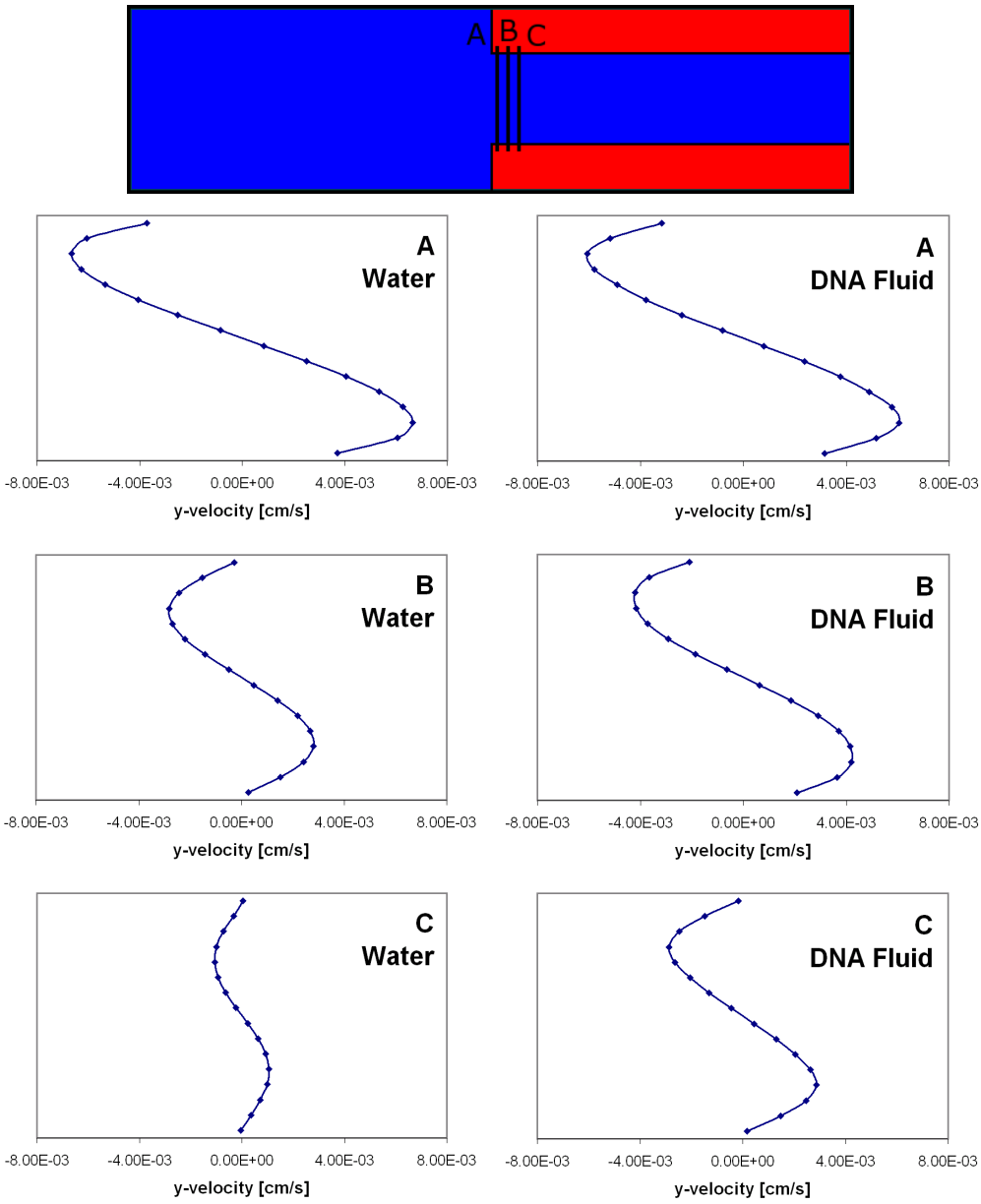

Figure 5.16: Computed transverse velocity profiles for distilled water and the DNA fluid at various locations downstream of the 2:1 contraction, as indicated in the top image. The solid lines represent best-fit curves for the blue data points. The results are consistent with Figure 5.14, in which the entrance lengths are greater for the DNA fluid.

N	Variable	Coarse Error	Fine Error	Order
1	u_0	1.82e-04	3.21e-05	2.50
1	u_1	9.34e-05	1.78e-05	2.39
1	p	3.45e-02	6.73e-03	2.36
2	u_0	5.70e-04	1.38e-04	2.05
2	u_1	2.24e-04	6.22e-05	1.85
2	p	3.60e-02	7.17e-03	2.33
∞	u_0	1.31e-02	6.51e-03	1.01
∞	u_1	6.38e-03	3.64e-03	0.81
∞	p	1.68e-01	4.06e-02	2.05

Table 5.7: Solution error convergence rates for distilled water past a sphere.

5.2.3 Flow Past a Sphere

The next test case is the flow of the experimental fluids past a sphere. The computational domain has $l = 0.052$ [cm] and $w = 0.013$ [cm] to match the width of the downstream experimental contraction. A sphere of radius 0.00195 [cm] is placed in the center of the channel. The initial velocity, stress, and pressure as well as the inflow velocity and stress are set to the analytic solution for fully developed Poiseuille flow in the absence of the sphere.

For distilled water, the coarse domain resolution is 128×32 and the coarse time step is 3.0×10^{-4}, corresponding to CFL ≈ 0.5. The steady-state velocity and pressure profiles are shown in Figure 5.17. The solution error convergence after 200 fine time steps is given in Table 5.7.

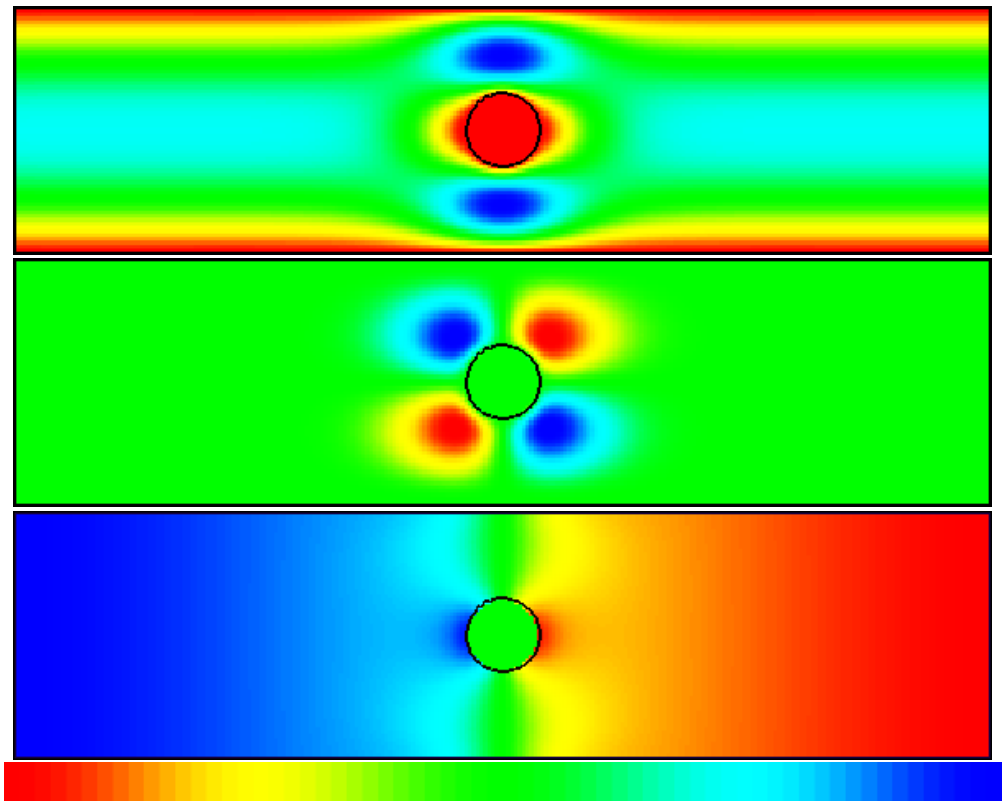

Figure 5.17: Steady-state profiles of distilled water past a sphere. From top to bottom: u_0, 0 (red) to 0.087 (blue) [cm/s]; u_1, -0.028 (red) to 0.028 (blue) [cm/s]; hydrostatic pressure p, 0 (red) to 3.47 (blue) [g/(cm-s^2)].

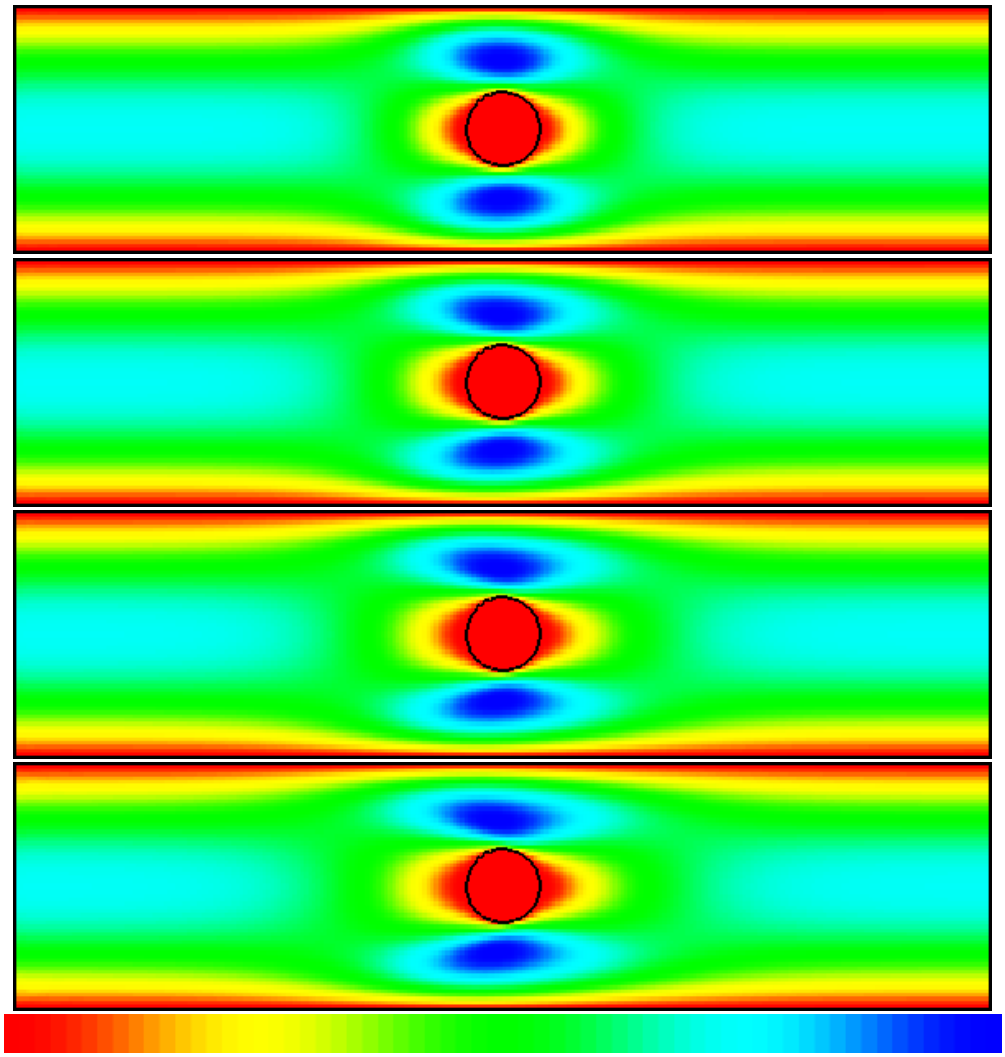

Figure 5.18: Time-dependent u_0 profiles of a DNA fluid past a sphere. The grid resolution is 256×64 with 200 time step increments using $\Delta t = 1.0 \times 10^{-5}$ [s]. Scale: 0 (red) to 0.087 (blue) [cm/s].

For the DNA fluid, the coarse domain resolution is 128×32 and the coarse time step is 2.0×10^{-5}, corresponding to CFL ≈ 0.5. The time-dependent normal velocity is shown in Figure 5.18. The transverse velocity, stress, and pressure corresponding to the final image of normal velocity are shown in Figure 5.19. The solution error convergence after 100 fine time steps is given in Table 5.8. Note that even for this geometry, increased entrance lengths occur past the sphere for the DNA fluid.

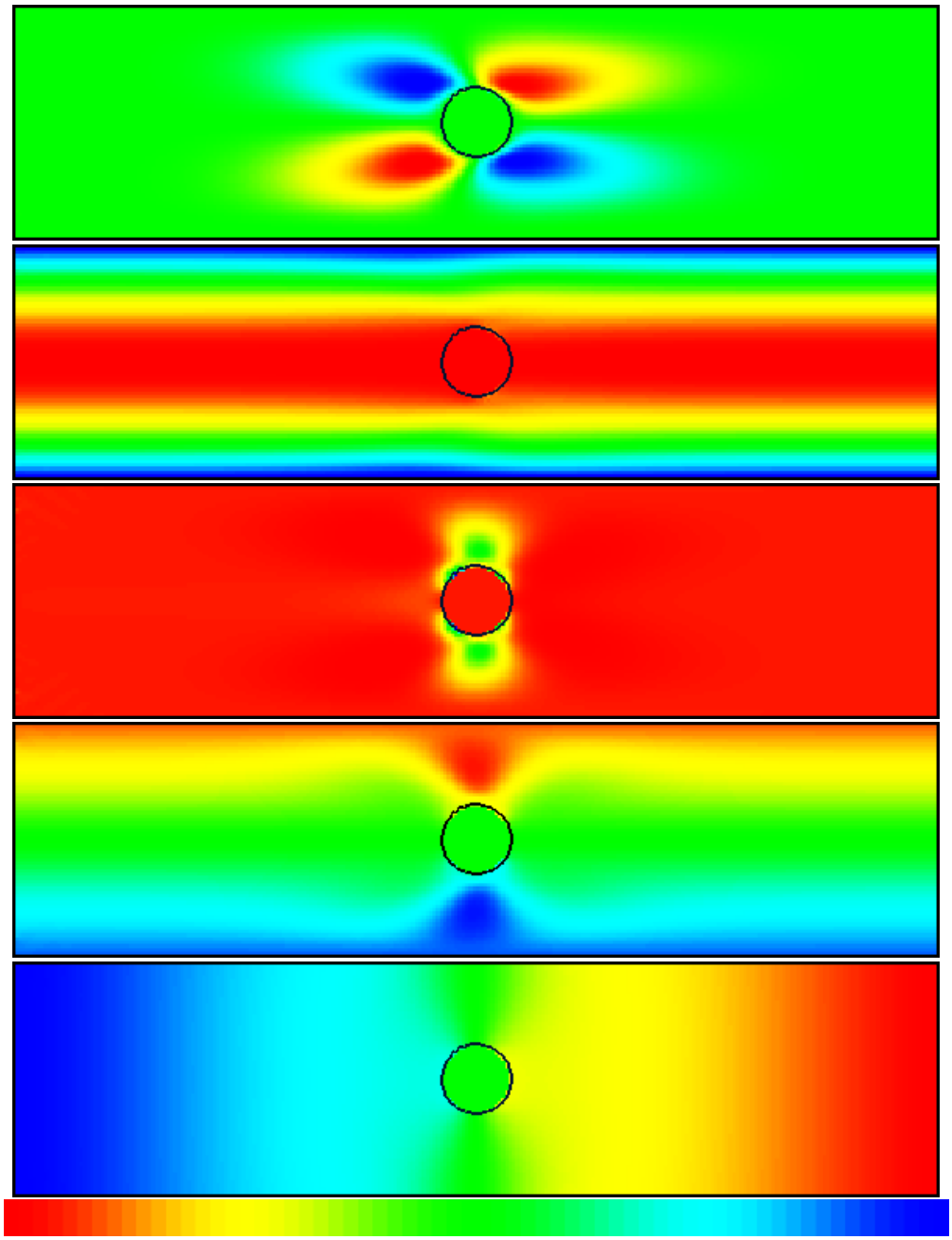

Figure 5.19: Profiles for a DNA fluid past a sphere at $t = 0.008$ [s] (corresponding to the last image in Figure 5.18). From top to bottom: u_1, -0.016 (red) to 0.016 (blue) [cm/s]; normal stress τ_{00}, -0.005 (red) to 40 (blue) [g/(cm-s^2)]; normal stress τ_{11}, -0.02 (red) to 0.38 (blue) [g/(cm-s^2)]; shear stress τ_{10}, -1.44 (red) to 1.44 (blue) [g/(cm-s^2)]; hydrostatic pressure p, 0 (red) to 13.8 (blue) [g/(cm-s^2)].

N	Variable	Coarse Error	Fine Error	Order
1	u_0	8.31e-05	1.97e-05	2.08
1	u_1	2.75e-05	7.66e-06	1.85
1	τ_{00}	9.96e-03	2.52e-03	1.98
1	τ_{10}	1.69e-04	4.24e-05	2.00
1	τ_{11}	1.77e-05	4.74e-06	1.90
1	p	2.11e-02	4.86e-03	2.11
2	u_0	3.10e-04	1.12e-04	1.47
2	u_1	1.16e-04	5.03e-05	1.20
2	τ_{00}	1.40e-02	4.35e-03	1.69
2	τ_{10}	6.94e-04	2.43e-04	1.52
2	τ_{11}	5.82e-05	2.03e-05	1.52
2	p	2.18e-02	5.06e-03	2.11
∞	u_0	1.25e-02	6.82e-03	0.87
∞	u_1	8.06e-03	3.75e-03	1.10
∞	τ_{00}	3.76e-01	1.99e-01	0.92
∞	τ_{10}	3.31e-02	1.17e-02	1.50
∞	τ_{11}	3.54e-03	1.24e-03	1.51
∞	p	6.79e-02	2.16e-02	1.65

Table 5.8: Solution error convergence rates for the DNA fluid past a sphere.

5.3 Conclusions

For each of the smooth test problems, the algorithm exhibits second-order solution error convergence in L^1 and first-order in L^∞ for velocity and stress with an advective CFL time step constraint of CFL ≈ 0.5, as expected. This is an improvement over [77], in which less than second order convergence was obtained with a smaller time step, and the algorithm did not support arbitrary smooth geometries. The algorithm also exhibits at least first-order convergence in L^1 for pressure, as expected. In some cases, such as the Maxwell fluid in the rectangular geometry, the convergence rates in L^∞ exceed first-order. This is due to the fact that given the position and shape of the expanded vortex, the largest magnitude errors occur in the interior of the domain, where the algorithm is second-order. For the abrupt contraction geometry, the algorithm also predicts the increased entrance lengths observed in real experimental fluids in microcontraction channels. This phenomena can be seen in the computational flow of experimental fluids past a sphere as well.

For this algorithm, a particular item requiring further study is the need for additional projection filters to smooth out the divergence in the velocity field in irregular cells. The problem requires additional filtering with increased resolution. Perhaps the solution is not a more robust filtering algorithm, but perhaps a new covered face state extrapolation that prevents these oscillations from occurring.

The first obvious extension to this work is a three-dimensional discretization of the equations. The upwind method [24] and discretizations for Poisson's equation and the heat equation [71] in a three-dimensional embedded boundary framework have already been developed, so the extension is straightforward. The methods in this thesis have been developed under the assumption that the geometry is sufficiently smooth. Additional studies are required to determine the robustness of the algorithm in the presence of discontinuous geometries, such as abrupt contractions. This will enable comparisons to standard benchmark problems [4, 65, 66, 77], such

as the flow of elastic liquids in hard-cornered planar and axisymmetric contractions. Additional studies are also required to robustness of this algorithm under higher values of We and Ma, and for a larger variety of experimental fluids and operating conditions [38, 39, 40]. In addition, adaptive numerical algorithms for the incompressible Navier-Stokes equations, in which the grid is locally refined in regions of interest, are being developed [53]. Adaptive techniques have already been used with success for hyperbolic conservation laws [24], so these two methods can be combined to develop a new adaptive projection method for incompressible viscoelasticity. Finally, another obvious extension is the discretization of more advanced constitutive models, such as those presented in Section 2.3. The methods in this thesis provide a framework for including the additional terms present in such models.

A Appendix

A.1 Difference Approximations

The reconstruction algorithm uses fourth-order van Leer limited differences [24, 80] to approximate the derivative. The notation $F = A \mid B \mid C$ means that formula A is used for F if the adjacent high and low side cells are open, the 2-point formula B is used if the high side cell is covered or outside of the domain, and the 2-point formula C is used if the low side cell is covered or outside of the domain. The difference approximation is defined as follows:

$$\begin{aligned}
\Delta_d^4 \mathbf{W_i} &= \Delta^{\mathrm{vL}}\left(\Delta_d^{4C}\mathbf{W_i}, \Delta_d^L \mathbf{W_i}, \Delta_d^R \mathbf{W_i}\right) \mid \Delta_d^L \mathbf{W_i} \mid \Delta_d^R \mathbf{W_i};\\
\Delta_d^{4C} \mathbf{W_i} &= \frac{2}{3}\left[\left(\mathbf{W} - \frac{1}{4}\Delta_d^2 \mathbf{W}\right)_{\mathbf{i}+\mathbf{e}_d} - \left(\mathbf{W} + \frac{1}{4}\Delta_d^2 \mathbf{W}\right)_{\mathbf{i}-\mathbf{e}_d}\right];\\
\Delta_d^2 \mathbf{W_i} &= \Delta^{\mathrm{vL}}\left(\Delta_d^{2C}\mathbf{W_i}, \Delta_d^L \mathbf{W_i}, \Delta_d^R W_{\mathbf{i}}\right) \mid \Delta_d^L \mathbf{W_i} \mid \Delta_d^R \mathbf{W_i};\\
\Delta_d^{2C} \mathbf{W_i} &= \frac{1}{2}\left(\mathbf{W}_{\mathbf{i}+\mathbf{e}_d} - \mathbf{W}_{\mathbf{i}-\mathbf{e}_d}\right),\\
\Delta_d^L \mathbf{W_i} &= \mathbf{W_i} - \mathbf{W}_{\mathbf{i}-\mathbf{e}_d},\\
\Delta_d^R \mathbf{W_i} &= \mathbf{W}_{\mathbf{i}+\mathbf{e}_d} - \mathbf{W_i},
\end{aligned} \tag{A.1}$$

where the van Leer limiting is applied to the primitive variables component-wise using:

$$\Delta^{\mathrm{vL}}(X, Y, Z) = \begin{cases} \mathrm{sign}(X) \cdot \min(|X|, 2|Y|, 2|Z|), & Y \cdot Z > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

The reconstruction algorithm also makes use of a second-order non-limited difference to approximate the derivative:

$$\mathcal{D}_d^2 \mathbf{W_i} = \Delta_d^{2C} \mathbf{W_i} \mid \Delta_d^L \mathbf{W_i} \mid \Delta_d^R \mathbf{W_i}. \tag{A.3}$$

A.2 Gradient Discretizations

There are three gradient discretizations used in this thesis, each operating on a scalar field ϕ. In situations where the gradient of a vector field is needed, the gradient of each component is computed separately. The first discretization is the cell-centered gradient of a cell-centered field, $(\nabla\phi)^{CC\to CC}$. This discretization relies completely on the results for the second discretization, which is the face-centered gradient of a cell-centered field, $(\nabla\phi)^{CC\to FC}$. To obtain the face-centered gradient at covered faces, there are three possible extrapolation procedures which will be described. For faces that correspond to domain boundaries, the discretizations for Dirichlet, homogeneous Neumann, and extrapolation boundary conditions will be described. The third gradient discretization is the EB centroid normal gradient of a cell-centered field, $(\mathbf{n}\cdot\nabla\phi)^{CC\to EB}$, which is always associated with either a Dirichlet or homogeneous Neumann boundary condition. The first two gradient discretizations return vector fields, while the third returns a scalar field.

A.2.1 Cell-Centered Gradient

The cell-centered gradient of a cell-centered field is defined as the average of the two normal face-centered gradients (see Appendix Section A.2.2), regardless of whether the face is open, covered, or a domain boundary:

$$(\nabla\phi)^{CC\to CC}_{d,\mathbf{i}} = \frac{1}{2}\left[(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} + (\nabla\phi)^{CC\to FC}_{d,\mathbf{i}-\frac{1}{2}\mathbf{e}_d}\right]. \tag{A.4}$$

For regular cells away from domain boundaries, this discretization reduces to the standard two-point centered difference approximation given in (4.83).

A.2.2 Face-Centered Gradient

For open faces, the face-centered normal gradient of a cell-centered field is the two-point difference approximation:

$$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{\phi_{\mathbf{i}+\mathbf{e}_d} - \phi_{\mathbf{i}}}{h}. \tag{A.5}$$

For open faces, the face-centered transverse gradient of a cell-centered field is the average of the two neighboring cell-centered transverse gradients:

$$(\nabla\phi)^{CC\to FC}_{d',\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \frac{1}{2}\left[(\nabla\phi)^{CC\to CC}_{d',\mathbf{i}} + (\nabla\phi)^{CC\to CC}_{d',\mathbf{i}+\mathbf{e}_d}\right]. \tag{A.6}$$

For covered faces and domain boundaries, these stencils must be modified since they require data points from outside of the solution domain. There are three options for obtaining $(\nabla\phi)^{CC\to FC}$ at covered faces: a geometrical extrapolation, a two-dimensional extrapolation, and a one-dimensional extrapolation. There are three possible boundary conditions for $(\nabla\phi)^{CC\to FC}$ at domain boundaries: Dirichlet, homogeneous Neumann, and extrapolation.

Geometrical extrapolation for the gradient at covered faces: The geometrical extrapolation for the gradient at covered faces only computes the normal gradient. This extrapolation is used for computing cell-centered gradients in the deformation tensor $2\mathbf{D}$, and the cell-centered $\nabla\phi$ in the cell-centered projection. In the geometrical extrapolation, a ray is cast into the domain from the covered face at the same orientation as the EB normal to obtain an extrapolation stencil using nearby face-centered gradients with the same orientation as the covered face, as illustrated in Figure A.1. The gradient at P is given by the following second-order extrapolation:

$$(\nabla\phi)_{d,P} = 2(\nabla\phi)_{d,Q} - (\nabla\phi)_{d,R}. \tag{A.7}$$

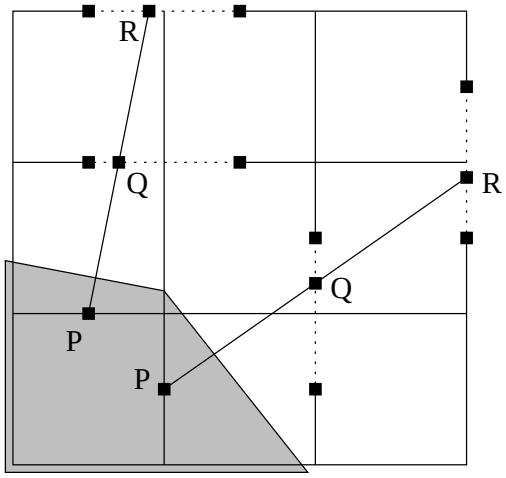

Figure A.1: Stencil points for the geometrical extrapolation for the gradient at covered faces. The covered face is at P. A ray is cast into the domain from P in the direction of the EB normal and obtain two values Q and R from neighboring faces with the same orientation as the covered face using linear interpolation from the two closest face-centers. A linear extrapolation is performed from points Q and R to get the value at P.

If any of the face-centered gradients needed in this stencil are covered or outside the domain, the algorithm switches to the two-dimensional extrapolation for the gradient at covered faces, defined next.

Two-dimensional extrapolation for the gradient at covered faces: The two-dimensional extrapolation for the gradient is only called if a stencil point for the geometrical extrapolation is covered or outside of the domain, and thus computes only the normal gradient. In the two-dimensional extrapolation, the highest order available approximation in the normal and transverse directions is used to obtain the gradient at covered face $A_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ (see Figure A.2). The gradient in the normal direction is defined as:

$$\textbf{if } \left(\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d\right) \text{ and } \left(\mathbf{i} \pm \frac{3}{2}\mathbf{e}_d\right) \text{ are both open } \textbf{then}$$
$$G_{N,d} = 2(\nabla\phi)_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}+\frac{3}{2}\mathbf{e}_d} \text{ and } \text{order}_N = 2$$

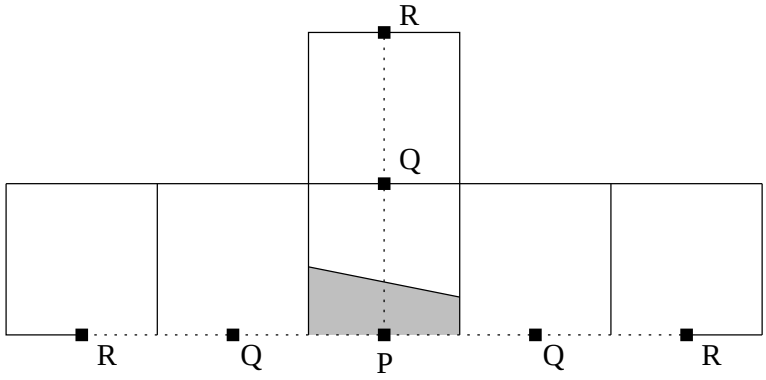

Figure A.2: Stencil points for the two-dimensional extrapolation for the gradient at covered faces. The covered face is at P. Gradients are linearly extrapolated from points Q and R (if they are open), and the highest order available stencil is used, using an averaging procedure in case of a tie.

else

$$G_{N,d} = (\nabla\phi)_{d,\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \text{ and } \text{order}_N = 1. \tag{A.8}$$

The gradient in the transverse direction is given by:

if $\left(\mathbf{i} + 2\mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$,$\left(\mathbf{i} + \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$,$\left(\mathbf{i} - \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$,
and $\left(\mathbf{i} - 2\mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ are all open **then**

$$G_{T,d} = \frac{1}{2}\Big[2(\nabla\phi)_{d,\mathbf{i}+\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}+2\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} + 2(\nabla\phi)_{d,\mathbf{i}-\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}-2\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d}\Big] \text{ and } \text{order}_T = 2$$

else if $\left(\mathbf{i} + 2\mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ and $\left(\mathbf{i} + \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ are both open **then**

$$G_{T,d} = 2(\nabla\phi)_{d,\mathbf{i}+\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}+2\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} \text{ and } \text{order}_T = 2$$

else if $\left(\mathbf{i} - 2\mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ and $\left(\mathbf{i} - \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ are both open **then**

$$G_{T,d} = 2(\nabla\phi)_{d,\mathbf{i}-\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}-2\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} \text{ and } \text{order}_T = 2$$

else if $\left(\mathbf{i} + \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ and $\left(\mathbf{i} - \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ are both open **then**

$$G_{T,d} = \frac{1}{2}\left[(\nabla\phi)_{d,\mathbf{i}+\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} + (\nabla\phi)_{d,\mathbf{i}-\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d}\right] \text{ and } \text{order}_T = 1$$

else if $\left(\mathbf{i} + \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right)$ is open **then**

$$
\begin{aligned}
&G_{T,d} = (\nabla\phi)_{d,\mathbf{i}+\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} \text{ and } \text{order}_T = 1 \\
&\textbf{else if } \left(\mathbf{i} - \mathbf{e}_{d'} \mp \frac{1}{2}\mathbf{e}_d\right) \text{ is open } \textbf{then} \\
&\quad G_{T,d} = (\nabla\phi)_{d,\mathbf{i}-\mathbf{e}_{d'}\mp\frac{1}{2}\mathbf{e}_d} \text{ and } \text{order}_T = 1 \\
&\textbf{else} \\
&\quad G_{T,d} = 0 \text{ and } \text{order}_T = 0.
\end{aligned} \tag{A.9}
$$

The gradient is then given by the highest order available approximation:

$$
\begin{aligned}
&\textbf{if } \text{ order}_N > \text{order}_T \textbf{ then} \\
&\quad (\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = G_{N,d} \\
&\textbf{else if } \text{ order}_T > \text{order}_N \textbf{ then} \\
&\quad (\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = G_{T,d} \\
&\textbf{else} \\
&\quad (\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \frac{1}{2}\left(G_{N,d} + G_{T,d}\right).
\end{aligned} \tag{A.10}
$$

One-dimensional extrapolation for the gradient at covered faces: Unlike the previous extrapolation procedures for the gradient at covered faces, the one-dimensional version computes both normal and transverse gradients. This extrapolation is used to compute both components of the face-centered $\nabla\phi$ in the MAC projection. In the one-dimensional extrapolation, the normal gradient at covered face $A_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ is found by extrapolating gradients in the normal direction (see Figure A.3):

$$
\begin{aligned}
&\textbf{if } \left(\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d\right) \text{ and } \left(\mathbf{i} \pm \frac{3}{2}\mathbf{e}_d\right) \text{ are both open } \textbf{then} \\
&\quad (\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 2(\nabla\phi)_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)_{d,\mathbf{i}\pm\frac{3}{2}\mathbf{e}_d} \\
&\textbf{else if } \left(\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d\right) \text{ is open } \textbf{then} \\
&\quad (\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = (\nabla\phi)_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} \\
&\textbf{else}
\end{aligned}
$$

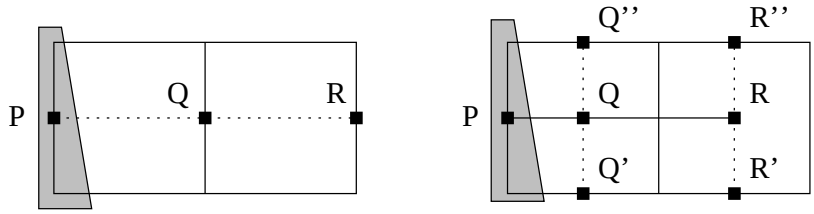

Figure A.3: Stencil points for the one-dimensional extrapolation for the gradient at covered faces. The covered face is at P. (Left) The normal gradient at P is a linear extrapolation of normal gradients from Q and R. (Right) The transverse gradient at P is a linear extrapolation of transverse gradients at Q and R, which are averages of the normal gradients at Q', Q'', R', and R''.

$$(\nabla\phi)_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 0. \tag{A.11}$$

The transverse gradient at covered face $A_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ is found with the following:

$$\begin{aligned}
&\textbf{if } (\mathbf{i}) \text{ and } (\mathbf{i} \pm \mathbf{e}_d) \text{ are both open } \textbf{then} \\
&\quad (\nabla\phi)_{d',\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \frac{3}{2}(\nabla\phi)_{d',\mathbf{i}} - \frac{1}{2}(\nabla\phi)_{d',\mathbf{i}\pm\mathbf{e}_d} \\
&\textbf{else} \\
&\quad (\nabla\phi)_{d',\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = (\nabla\phi)_{d',\mathbf{i}},
\end{aligned} \tag{A.12}$$

where the cell-centered transverse gradients needed in (A.12) are defined as:

$$\begin{aligned}
&\textbf{if } \left(\mathbf{i} + \frac{1}{2}\mathbf{e}_{d'}\right) \text{ and } \left(\mathbf{i} - \frac{1}{2}\mathbf{e}_{d'}\right) \text{ are both open } \textbf{then} \\
&\quad (\nabla\phi)_{d',\mathbf{i}} = \frac{1}{2}\left[(\nabla\phi)_{d',\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} + (\nabla\phi)_{d',\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}\right] \\
&\textbf{else if } \left(\mathbf{i} + \frac{1}{2}\mathbf{e}_{d'}\right) \text{ and } \left(\mathbf{i} + \frac{3}{2}\mathbf{e}_{d'}\right) \text{ are both open } \textbf{then} \\
&\quad (\nabla\phi)_{d',\mathbf{i}} = \frac{3}{2}(\nabla\phi)_{d',\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}} - \frac{1}{2}(\nabla\phi)_{d',\mathbf{i}+\frac{3}{2}\mathbf{e}_{d'}} \\
&\textbf{else if } \left(\mathbf{i} - \frac{1}{2}\mathbf{e}_{d'}\right) \text{ and } \left(\mathbf{i} - \frac{3}{2}\mathbf{e}_{d'}\right) \text{ are both open } \textbf{then} \\
&\quad (\nabla\phi)_{d',\mathbf{i}} = \frac{3}{2}(\nabla\phi)_{d',\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}} - \frac{1}{2}(\nabla\phi)_{d',\mathbf{i}-\frac{3}{2}\mathbf{e}_{d'}} \\
&\textbf{else if } \left(\mathbf{i} + \frac{1}{2}\mathbf{e}_{d'}\right) \text{ is open } \textbf{then} \\
&\quad (\nabla\phi)_{d',\mathbf{i}} = (\nabla\phi)_{d',\mathbf{i}+\frac{1}{2}\mathbf{e}_{d'}}
\end{aligned}$$

else if $\left(\mathbf{i} - \frac{1}{2}\mathbf{e}_{d'}\right)$ is open **then**

$(\nabla\phi)_{d',\mathbf{i}} = (\nabla\phi)_{d',\mathbf{i}-\frac{1}{2}\mathbf{e}_{d'}}$

else

$(\nabla\phi)_{d',\mathbf{i}} = 0.$ (A.13)

Domain boundary conditions for the face-centered gradient: There are three types of boundary conditions used for obtaining $(\nabla\phi)^{CC\to FC}$ at domain boundaries: Dirichlet, homogeneous Neumann, and extrapolation. The domain boundary face is at $A_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ and only the normal gradients are needed. For the Dirichlet condition $\phi_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = f$, the following first-order difference approximation is used:

$$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \mp\frac{f - \phi_{\mathbf{i}}}{0.5h}. \tag{A.14}$$

For the homogeneous Neumann condition $(\mathbf{n}\cdot\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 0$, the exact condition can be imposed:

$$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 0. \tag{A.15}$$

For an extrapolation boundary condition, linear extrapolation is used:

if $\left(\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d\right)$ and $\left(\mathbf{i} \pm \frac{3}{2}\mathbf{e}_d\right)$ are both open **then**

$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 2(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} - (\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\pm\frac{3}{2}\mathbf{e}_d}$

else if $\left(\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d\right)$ is open **then**

$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = (\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$

else

$(\nabla\phi)^{CC\to FC}_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = 0.$ (A.16)

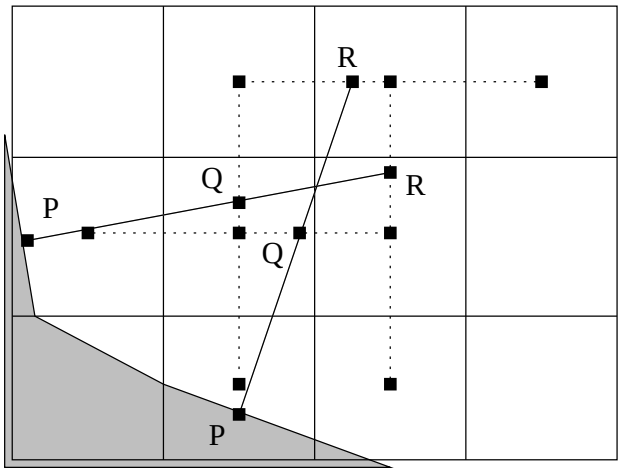

Figure A.4: Stencil points for the EB centroid normal gradient. The EB centroid is at point P. A ray is cast into the domain from P in the direction of the EB normal and obtain two values ϕ_Q and ϕ_R from neighboring cell columns (if $\theta < \pi/4$) or rows (if $\pi/4 \leq \theta \leq \pi/2$) using quadratic polynomial interpolation from the three nearest cell-centers. Here, θ is the magnitude of the smallest angle between the ray and the horizontal. Boundary conditions are applied and a quadratic polynomial approximation is used to obtain the normal gradient at P.

A.2.3 EB Centroid Normal Gradient

To compute $(\mathbf{n} \cdot \nabla\phi)^{CC \to EB}$, there are two possible boundary conditions at the EB. A homogeneous Neumann condition $(\mathbf{n} \cdot \nabla\phi)^{CC \to EB} = 0$ specifies the gradient exactly and no computations are required. For a Dirichlet condition, $\phi_B = f$, the normal gradient is computed using the second-order extrapolation technique described in [42], in which a ray is cast into the domain from the EB centroid in the direction of the EB normal to obtain an extrapolation stencil, as illustrated in Figure A.4. The normal gradient at P is computed by taking the derivative of a quadratic polynomial approximation to ϕ [16]:

$$(\mathbf{n} \cdot \nabla\phi)_P = \frac{l_1^2 (\phi_R - \phi_P) - l_2^2 (\phi_Q - \phi_P)}{l_1 l_2 (l_2 - l_1)}; \qquad l_1 = |\overline{PQ}|, \qquad l_2 = |\overline{PR}|. \tag{A.17}$$

A.3 Covered Face State Extrapolation

After the normal predictor Riemann problem, the one-dimensional MAC state, $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, is available at each open face and after the transverse predictor Riemann problem, the preliminary MAC state, $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$, is available at each open face. After each of these steps, the corresponding MAC state at covered faces is determined by extrapolating neighboring MAC states. The extrapolation is based on the technique described in [24], in which the inputs to open face Riemann problems are extrapolated to obtain a "covered-side" state. The "interior" face-centered, time-centered state is computed using the same Taylor series extrapolation procedure used for open faces in the normal and transverse predictor steps. A covered-face Riemann problem is solved using the interior and covered-side states as inputs. In contrast, the approach taken here is to extrapolate the MAC state. In this work, it has been observed that this form of covered face extrapolation provides additional stability in irregular cells, in particular cells with small volume fractions.

The details of the extrapolation are now described for the one-dimensional MAC state. The extrapolation for the preliminary MAC state is identical, except that $\mathbf{W}^{n+\frac{1}{2},*}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ is used as the input state rather than $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$. In particular, if cell $\mathbf{i}$ is open and cell $\mathbf{i}\mp\mathbf{e}_d$ is covered then the covered face value $\mathbf{W}^{1D}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ is extrapolated from a collection of values $\mathbf{W}^{1D}_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ that have been previously computed at each open face. First, the control volumes involved are defined as:

$$\begin{aligned}
\mathbf{i}_u &= \mathbf{i} + s_{d'}\mathbf{e}_{d'} - s_d\mathbf{e}_d, \\
\mathbf{i}_s &= \mathbf{i} + s_d\mathbf{e}_d, \\
\mathbf{i}_c &= \mathbf{i} + s_{d'}\mathbf{e}_{d'},
\end{aligned} \tag{A.18}$$

where $d' \neq d$ and $s_d = \text{sign}(n_d)$, where here $\mathbf{n}$ is the inward normal. Next, the

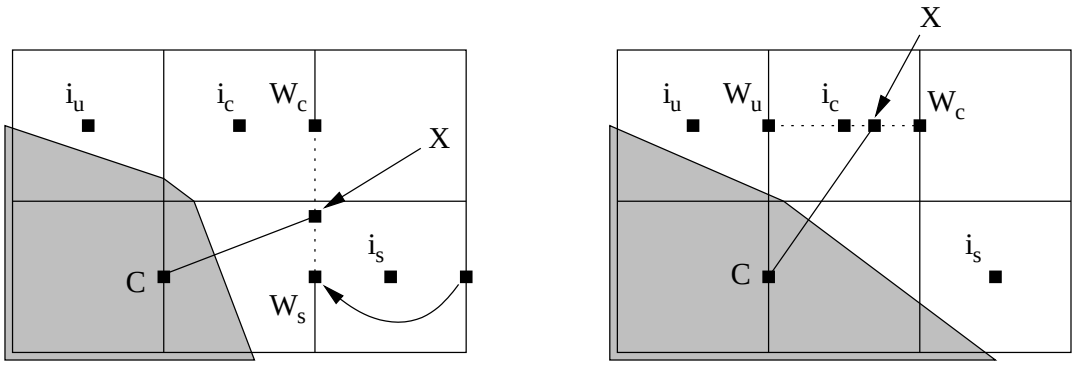

Figure A.5: Extrapolation of the one-dimensional (and preliminary) MAC state to covered faces. The covered face is at C. (Case 1, Left) Extrapolation is used to obtain $\mathbf{W}_s$ and then the value at X is determined using linear interpolation between $\mathbf{W}_s$ and $\mathbf{W}_c$ where the boundary normal intersects the dashed line. An extrapolation procedure along the normal is used to get the value at the covered face. (Case 2, Right) An interpolation procedure is used between $\mathbf{W}_u$ and $\mathbf{W}_c$ to X where the boundary normal intersects the dashed line. An extrapolation procedure is used back along the normal to the covered face.

extrapolations of the state variables to nearby edges are defined as:

$$\begin{aligned} \mathbf{W}_u &= \mathbf{W}^{1D}_{\mathbf{i}_u \pm \frac{1}{2}\mathbf{e}_d}, \\ \mathbf{W}_s &= \mathbf{W}^{1D}_{\mathbf{i}_s \pm \frac{1}{2}\mathbf{e}_d} - s_d \Delta_d^2 \mathbf{W}^n_{\mathbf{i}_s}, \\ \mathbf{W}_c &= \mathbf{W}^{1D}_{\mathbf{i}_c \pm \frac{1}{2}\mathbf{e}_d}. \end{aligned} \tag{A.19}$$

To extrapolate to the covered faces, a linear combination of the values defined above are used to compute the value along a ray normal to the EB and passing through the center of the covered face. The values are then extrapolated to the covered face using second-order limited slopes. The details are as follows:

Case 1, illustrated in Figure A.5, left: If $|n_d| \geq |n_{d'}|$, then

$$\mathbf{W}^{1D}_{\mathbf{i} \pm \frac{1}{2}\mathbf{e}_d} = \frac{|n_{d'}|}{|n_d|}\mathbf{W}_c + \left(1 - \frac{|n_{d'}|}{|n_d|}\right)\mathbf{W}_s - \left(\frac{|n_{d'}|}{|n_d|} s_{d'} \Delta_{d'} \mathbf{W} + s_d \Delta_d \mathbf{W}\right); \tag{A.20}$$

$$\Delta_{d''}\mathbf{W} = \frac{n_{d'}}{n_d}\Delta^2_{d''}\mathbf{W}^n_{\mathbf{i}_c} + \left(1 - \frac{|n_{d'}|}{|n_d|}\right)\Delta^2_{d''}\mathbf{W}^n_{\mathbf{i}_s}. \tag{A.21}$$

Case 2, in Figure A.5, Right: If $|n_d| < |n_{d'}|$, then

$$\mathbf{W}^{1D}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \frac{|n_d|}{|n_{d'}|}\mathbf{W}_c + \left(1 - \frac{|n_d|}{|n_{d'}|}\right)\mathbf{W}_u - \left(\frac{|n_d|}{|n_{d'}|}s_d\Delta_d\mathbf{W} + s_{d'}\Delta_{d'}\mathbf{W}\right); \quad \text{(A.22)}$$

$$\Delta_{d''}\mathbf{W} = \frac{n_d}{n_{d'}}\Delta^2_{d''}\mathbf{W}^n_{\mathbf{i}_c} + \left(1 - \frac{|n_d|}{|n_{d'}|}\right)\Delta^2_{d''}\mathbf{W}^n_{\mathbf{i}_u}. \quad \text{(A.23)}$$

If one or both values of $\mathbf{W}_u$, $\mathbf{W}_c$, or $\mathbf{W}_s$ used in the extrapolation lie on a covered face, the algorithm drops order. First, if $\mathbf{i}_s$ is open, then $\mathbf{W}^{1D}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \mathbf{W}^{1D}_{\mathbf{i}_s\pm\frac{1}{2}\mathbf{e}_d} - 2s_d\Delta^2_d\mathbf{W}^n_{\mathbf{i}_s}$. Otherwise, if exactly one of the values of $\mathbf{W}_u$, $\mathbf{W}_c$, or $\mathbf{W}_s$ used in the extrapolation is covered, then $\mathbf{W}^{1D}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ is set to the value corresponding to the open face. If both faces are covered, then $\mathbf{W}^{1D}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \mathbf{W}^n_{\mathbf{i}}$.

A.4 Multigrid Solver

The discretizations for the diffusion equation Helmholtz operators (4.131) and (4.132), the MAC projection Poisson equation (4.149), and the cell-centered projection Poisson equation 4.157) can be represented as a system of n linear equations, where n is the total number of control volumes in the domain. Each of these systems of linear equations can be represented as a matrix equations of the form:

$$\begin{bmatrix} & & \\ & \boldsymbol{\mathcal{L}} & \\ & & \end{bmatrix} \cdot \begin{bmatrix} \phi_{ij} \\ \phi_{i+1,j} \\ \vdots \\ \phi_{i+i_{\max},j} \\ \phi_{i,j+1} \\ \vdots \\ \phi_{i+i_{\max},j+j_{\max}} \end{bmatrix} = \begin{bmatrix} b_{ij} \\ b_{i+1,j} \\ \vdots \\ b_{i+i_{\max},j} \\ b_{i,j+1} \\ \vdots \\ b_{i+i_{\max},j+j_{\max}} \end{bmatrix}. \quad \text{(A.24)}$$

Ignoring the effects of round-off error, these systems can be solved exactly using direct methods such as Gaussian elimination, but these methods are computationally

expensive, requiring $\mathcal{O}(n^3)$ operations. For modest sized problems where $n = 512^2$ (for a 512×512 domain), the total number of operations required to solve the linear system directly is approximately $\mathcal{O}(512^6) \approx 1.8 \times 10^{16}$. Obviously, it is not practical to use direct solvers for problems of this size, so iterative methods are used.

Consider an elliptic equation of the form (4.46), or a parabolic equation of the form (4.47). These equations can be represented in matrix form:

$$\boldsymbol{\mathcal{L}}\phi = b. \tag{A.25}$$

An iterative method repeatedly updates an approximation to the solution, $\hat{\phi}$, until an acceptable solution is reached, as measured by a norm of the residual, ρ:

$$\rho = b - \boldsymbol{\mathcal{L}}\hat{\phi}. \tag{A.26}$$

Performing an iterative update on the approximate solution is often referred to as "relaxing" the approximate solution. Using an iterative method, the norm of the residual can become acceptably small using significantly less operations than a direct method. A basic relaxation method is Jacobi relaxation, which takes the form:

$$\phi_{\mathbf{i}}^{k+1} = \phi_{\mathbf{i}}^{k} + \omega_{\mathbf{i}} \left[b_{\mathbf{i}} - (\boldsymbol{\mathcal{L}}\phi^k)_{\mathbf{i}} \right], \tag{A.27}$$

where the superscript is the iteration index and $\omega_{\mathbf{i}}$ is a problem-dependent weighting coefficient. Jacobi relaxation is suitable for parallel processing since the solution approximations for each control volume are updated simultaneously.

Another relaxation method is Gauss-Seidel relaxation, which takes the form:

$$\phi_{\mathbf{i}}^{k+1} = \phi_{\mathbf{i}}^{k} + \omega_{\mathbf{i}} \left[b_{\mathbf{i}} - (\boldsymbol{\mathcal{L}}\tilde{\phi}^k)_{\mathbf{i}} \right], \tag{A.28}$$

where the solution approximations for each control volume are updated sequen-

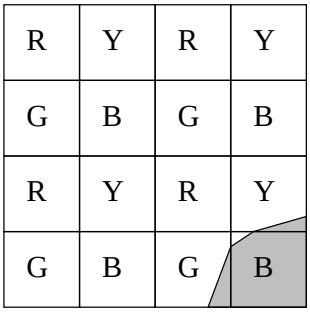

Figure A.6: In a multicolored Gauss-Seidel relaxation, the cells are labeled as either red (R), yellow (Y), green (G), or black (B) in the following pattern, regardless of whether a cell is regular or irregular.

tially and the auxiliary approximation $\tilde{\phi}^k$ is the original approximation ϕ^k that is overwritten with values of $\phi_{\mathbf{i}}^{k+1}$ as they become available. Gauss-Seidel relaxation often reduces the norm of the residual more quickly than Jacobi relaxation, but is not suitable for parallel processing since the solutions for each control volume are updated sequentially.

A third relaxation method is red-black Gauss-Seidel relaxation, named after the red-black pattern that appears on checkerboards. Each cell is alternately labeled as either "red" or "black". The red cells are updated with a Jacobi relaxation. Next the black cells are updated with a Jacobi relaxation using an auxiliary approximation $\tilde{\phi}^k$, which consists of updated approximations ϕ^{k+1} from the red cells and the original approximations ϕ^k from the black cells. Red-black Gauss-Seidel relaxation is suitable for parallel processing since groups of solution approximations are updated simultaneously.

The relaxation method used in this thesis is a multicolored Gauss-Seidel relaxation, which is a variant of red-black Gauss-Seidel relaxation. All regular cells are labeled as either red, yellow, green, or black in an alternating fashion (see Figure A.6). The cells are relaxed in the following sweep order: regular red, irregular red, regular yellow, irregular yellow, regular green, irregular green, regular blue, and ir-

regular blue. Similar to red-black Gauss-Seidel, after each set of cells are relaxed, the subsequent relaxations in the sweep use the most recently computed values. The weighting coefficient $\omega_{\mathbf{i}}$ is computed for each cell by taking the Laplacian of the following "delta" function:

$$\frac{1}{\omega_{\mathbf{i}}} = \boldsymbol{\mathcal{L}}\delta_{\mathbf{i}}, \qquad \delta_{\mathbf{i}} = \begin{cases} 1, & \text{in cell } \mathbf{i}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.29}$$

Note that using this formulation, the EB face does not contribute to ω, since the gradient stencil at the EB face does not include the cell-centered value, and therefore is equal to zero.

One of the disadvantages to iterative methods is that low frequency error modes are damped very slowly, requiring a very large number of iterations to approach the exact solution. Multigrid is an iterative technique in which a specified number of iterations are performed, and then the residual is transferred to a coarser grid using a restriction operator. An iterative procedure is performed on the coarse grid to determine a correction that will be used to correct the fine grid approximation. The concept behind multigrid is that low frequency errors that are slowly damped using iterative methods on a fine grid will appear as higher frequency errors when transferred to a coarser grid. Refer to [15] for a detailed description of the multigrid implementation described here. The details for a two-level multigrid scheme are as follows:

1. The goal is to iteratively solve:

$$\boldsymbol{\mathcal{L}}_h\phi_h = b_h. \tag{A.30}$$

Choose an initial guess, $\hat{\phi}_h$, and relax the fine grid solution ν_1 times.

2. Compute the fine grid residual, ρ_h:

$$\rho_h = b_h - \boldsymbol{\mathcal{L}}_h \hat{\phi}_h. \tag{A.31}$$

3. Use a restriction operator, I_h^{2h}, to transfer the fine grid residual to a coarse grid:

$$b_{2h} = I_h^{2h} \rho_h. \tag{A.32}$$

4. Now the goal is to iteratively solve the residual equation:

$$\boldsymbol{\mathcal{L}}_{2h} \phi_{2h} = b_{2h}, \tag{A.33}$$

with homogeneous Dirichlet boundary conditions. Choose an initial guess, $\hat{\phi}_{2h} = 0$, and relax the coarse grid solution ν_1 times.

5. Use an interpolation operator, I_{2h}^h, to correct the fine grid solution with the coarse grid solution:

$$\hat{\phi}_h \leftarrow \hat{\phi}_h + I_{2h}^h \hat{\phi}_{2h}. \tag{A.34}$$

6. Relax the fine grid solution ν_2 times.

A more elaborate scheme used in this thesis involves successively transferring the residuals from each resolution to a coarser grid. A full step is referred to as a V-cycle, and the boundary conditions for all the residual problems are homogeneous Dirichlet. The method is given as follows:

- Relax on $\boldsymbol{\mathcal{L}}_h \phi_h = b_h$ ν_1 times with an arbitrary initial guess $\hat{\phi}_h$.
- Compute $b_{2h} = I_h^{2h}(b_h - \boldsymbol{\mathcal{L}}_h \hat{\phi}_h)$.
    - Relax on $\boldsymbol{\mathcal{L}}_{2h} \phi_{2h} = b_{2h}$ ν_1 times with an initial guess $\hat{\phi}_{2h} = 0$.
    - Compute $b_{4h} = I_{2h}^{4h}(b_{2h} - \boldsymbol{\mathcal{L}}_{2h} \hat{\phi}_{2h})$.

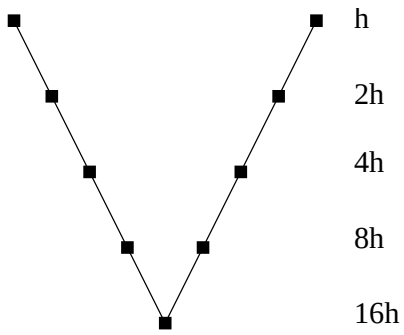

Figure A.7: V-cycle for four levels of refinement. The coarsest mesh has a grid spacing of $16h$.

- Relax on $\boldsymbol{\mathcal{L}}_{4h}\phi_{4h} = b_{4h}$ ν_1 times with an initial guess $\hat{\phi}_{4h} = 0$.
- Compute $b_{8h} = I_{4h}^{8h}(b_{4h} - \boldsymbol{\mathcal{L}}_{4h}\hat{\phi}_{4h})$.

  $\vdots$

  - Solve $\boldsymbol{\mathcal{L}}_{Lh}\hat{\phi}_{Lh} = b_{Lh}$.

  $\vdots$
- Correct $\hat{\phi}_{4h} \leftarrow \hat{\phi}_{4h} + I_{8h}^{4h}\hat{\phi}_{8h}$.
- Relax on $\boldsymbol{\mathcal{L}}_{4h}\phi_{4h} = b_{4h}$ ν_2 times with an initial guess $\hat{\phi}_{4h}$.

- Correct $\hat{\phi}_{2h} \leftarrow \hat{\phi}_{2h} + I_{4h}^{2h}\hat{\phi}_{4h}$.
- Relax on $\boldsymbol{\mathcal{L}}_{2h}\phi_{2h} = b_{2h}$ ν_2 times with an initial guess $\hat{\phi}_{2h}$.

- Correct $\hat{\phi}_h \leftarrow \hat{\phi}_h + I_{2h}^{h}\hat{\phi}_{2h}$.
- Relax on $\boldsymbol{\mathcal{L}}_h\phi_h = b_h$ ν_2 times with an initial guess $\hat{\phi}_h$.

In this thesis, $\nu_1 = \nu_2 = 4$. The fine geometry is coarsened such that the area fractions and volume fractions are conserved. Note that using this representation, the EB cannot be represented as a line on coarser geometries. The residuals are transferred to a coarser grid using a volume weighted average. The corrections are interpolated to the finer grid using piecewise constant interpolation. After each V-cycle, the norm of the residual is computed. The iterative process is complete when

the max norm of the residual decreases by a factor of 10^{-12}. Before each multigrid V-cycle, a preconditioning step is used in which the finest solution is relaxed using 4 Jacobi relaxations. At the coarsest level, the BiConjugate Gradient Stabilized (BiCGSTAB) [7] iterative method is used to solve the problem.

A.5 Cell-Centered Projection Filter

The cell-centered projection filter requires a cell-centered discretization of $\nabla(\nabla \cdot \mathbf{u})$. Following [5], there are three steps in computing $\nabla(\nabla \cdot \mathbf{u})$. First, the cell-centered velocity gradients are computed from cell-centered data using:

$$
\begin{aligned}
&\textbf{if } (\mathbf{i} + \mathbf{e}_d) \text{ and } (\mathbf{i} - \mathbf{e}_d) \text{ are both open } \textbf{then} \\
&\quad (\nabla \mathbf{u})_{\mathbf{i}} = \frac{\mathbf{u}_{\mathbf{i}+\mathbf{e}_d} - \mathbf{u}_{\mathbf{i}-\mathbf{e}_d}}{2h} \\
&\textbf{else if } (\mathbf{i} \pm \mathbf{e}_d) \text{ and } (\mathbf{i} \pm 2\mathbf{e}_d) \text{ are both open } \textbf{then} \\
&\quad (\nabla \mathbf{u})_{\mathbf{i}} = \pm \left[\frac{4\left(\mathbf{u}_{\mathbf{i}\pm\mathbf{e}_d} - \mathbf{u}_{\mathbf{i}}\right) - \left(\mathbf{u}_{\mathbf{i}\pm 2\mathbf{e}_d} - \mathbf{u}_{\mathbf{i}}\right)}{2h} \right] \\
&\textbf{else if } (\mathbf{i} \pm \mathbf{e}_d) \text{ is open } \textbf{then} \\
&\quad (\nabla \mathbf{u})_{\mathbf{i}} = \pm \left[\frac{\mathbf{u}_{\mathbf{i}\pm\mathbf{e}_d} - \mathbf{u}_{\mathbf{i}}}{h} \right] \\
&\textbf{else} \\
&\quad (\nabla \mathbf{u})_{\mathbf{i}} = 0.
\end{aligned}
\tag{A.35}
$$

Next, the face-centered divergence is computed at non-domain faces as follows:

$$
(\nabla \cdot \mathbf{u})_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = \frac{u_{d,\mathbf{i}+\mathbf{e}_d} - u_{d,\mathbf{i}}}{h} + \frac{1}{2}\left[(\nabla \mathbf{u})_{d',\mathbf{i}+\mathbf{e}_d} + (\nabla \mathbf{u})_{d',\mathbf{i}}\right]. \tag{A.36}
$$

At domain boundaries, the following extrapolation procedure is used:

$$
\begin{aligned}
&\textbf{if } (\mathbf{i} \pm \mathbf{e}_d) \text{ is open } \textbf{then} \\
&\quad (\nabla \cdot \mathbf{u})_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d} = \pm \left[\frac{2.25\left(u_{d,\mathbf{i}} - u_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}\right) - 0.25\left(u_{d,\mathbf{i}\pm\mathbf{e}_d} - u_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}\right)}{0.75h} \right]
\end{aligned}
$$

$$+\left[\frac{3(\nabla\mathbf{u})_{d',\mathbf{i}}-(\nabla\mathbf{u})_{d',\mathbf{i}\pm\mathbf{e}_d}}{2}\right]$$

else

$$(\nabla\cdot\mathbf{u})_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}=\pm\left[\frac{u_{d,\mathbf{i}}-u_{d,\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}}{0.5h}\right]+(\nabla\mathbf{u})_{d',\mathbf{i}}, \tag{A.37}$$

where $\mathbf{u}_{\mathbf{i}\mp\frac{1}{2}\mathbf{e}_d}$ matches the domain boundary conditions in the cell-centered projection. Covered face divergences are obtained using the same geometrical extrapolation stencil used to compute normal gradients at covered faces. The last step is to calculate the non-conservative face-centered gradient of the face-centered divergence:

$$\left[\nabla(\nabla\cdot\mathbf{u})_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}\right]_{d,\mathbf{i}}=\frac{(\nabla\cdot\mathbf{u})_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}-(\nabla\cdot\mathbf{u})_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d}}{h}. \tag{A.38}$$

References

[1] M. Aboubacar, H. Matallah, and M. F. Webster. Highly elastic solutions for Oldroyd-B and Phan-Thien/Tanner fluids with a finite volume/element method: planar contraction flows. *J. Non-Newtonian Fluid Mech.*, 103:65–103, 2002.

[2] A. S. Almgren, J. B. Bell, and W. Y. Crutchfield. Approximate projection methods: Part I. Inviscid analysis. *J. Sci. Comput.*, 22:1139–1159, 2000.

[3] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *J. Sci. Comput.*, 17:358–369, 1996.

[4] M. A. Alves, P. J. Oliveira, and F. T. Pinho. Benchmark solutions for the flow of Oldroyd-B and PTT fluids in planar contractions. *J. Non-Newtonian Fluid Mech.*, 110:45–75, 2003.

[5] M. Barad, P. Colella, D. T. Graves, T. Ligocki, P. Schwartz, B. van Straalen, and D. Trebotich. A Cartesian grid embedded boundary method for the incompressible Navier-Stokes equations. In preparation, 2007.

[6] H. A. Barnes, J. F. Hutton, and K. Walters. *An Introduction to Rheology.* Elsevier, New York, 1989.

[7] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.

[8] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 85:257–283, 1989.

[9] M. J. Berger and R. J. Leveque. Stable boundary conditions for Cartesian grid calculations. Technical Report 90-37, ICASE, May 1990.

[10] D. M. Binding. Further considerations of axisymmetric contraction flows. *J. Non-Newtonian Fluid Mech.*, 41:27–42, 1991.

[11] D. M. Binding and K. Walters. On the use of flow through a contraction in estimating the extensional viscosity of mobile polymer solutions. *J. Non-Newtonian Fluid Mech.*, 30:233–250, 1988.

[12] R. B. Bird, R. C. Armstrong, and O. Hassager. *Dynamics of Polymeric Liquids.* John Wiley & Sons, New York, 1987.

[13] D. V. Boger. A highly elastic constant-viscosity fluid. *J. Non-Newtonian Fluid Mech.*, 3:87–91, 1977.

[14] D. V. Boger and K. Walters. *Rheological Phenomena in Focus.* Elsevier, New York, 1993.

[15] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial.* SIAM, Philadelphia, PA, 2000.

[16] S. C. Chapra and R. P. Canale. *Numerical Methods for Engineers: With Software and Programming Applications.* McGraw-Hill, New York, 2001.

[17] A. J. Chorin. Numerical solutions of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[18] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer-Verlag, New York, 1993.

[19] W. J. Coirier and K. G. Powell. An assessment of cartesian-mesh approaches for the euler equations. *J. Comp. Phys.*, 147:121–131, 1995.

[20] P. Colella. Glimm's method for gas dynamics. *J. Sci. Stat. Comput.*, 3:76–110, 1982.

[21] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *J. Sci. Stat. Comput.*, 6:104–117, 1985.

[22] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comp. Phys.*, 87:171–200, 1990.

[23] P. Colella. Volume-of-fluid methods for partial differential equations. In E. F. Toro, editor, *Godunov Methods: Theory and Applications.* Kluwer, New York, 2001.

[24] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *J. Comp. Phys.*, 211:347–366, 2006.

[25] P. Colella and P. R. Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comp. Phys.*, 54:174–201, 1984.

[26] B. D. Coleman and W. Noll. An approximation theorem for functionals, with applications in continuum mechanics. *Arch. Rat. Mech. Anal.*, 6:355–370, 1960.

[27] M. J. Crochet, A. R. Davies, and K. Walters. *Numerical Simulation of Non-Newtonian Flow.* Elsevier, New York, 1984.

[28] M. J. Crochet and K. Walters. Numerical methods in non-Newtonian fluid mechanics. *Ann. Rev. Fluid Mech.*, 15:241–260, 1983.

[29] R. K. Crockett, P. Colella, R. T. Fisher, R. I. Klein, and C. F. McKee. An unsplit, cell-centered Godunov method for ideal MHD. *J. Comp. Phys.*, 203:422–448, 2005.

[30] W. M. Deen. *Analysis of Transport Phenomena*. Oxford University Press, New York, 1998.

[31] R. Fattal and R. Kupferman. Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation. *J. Non-Newtonian Fluid Mech.*, 126:23–37, 2005.

[32] J.-M. Ghidaglia and F. Pascal. The normal flux method at the boundary for multidimensional finite volume approximations in CFD. *Eur. J. Mech. B Fluids*, 24:1–17, 2005.

[33] H. Giesekus. A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility. *J. Non-Newtonian Fluid Mech.*, 11:69–109, 1982.

[34] H. Giesekus. On configuration-dependent generalized Oldroyd derivatives. *J. Non-Newtonian Fluid Mech.*, 14:47–65, 1984.

[35] D. Gilbarg and N. S. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Springer, New York, 1977.

[36] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Math. Sbornik*, 47:271–306, 1959.

[37] S. K. Godunov, A. V. Zabrodin, and G. P. Prokopov. A computational scheme for two-dimensional non stationary problems of gas dynamics and calculation of the flow from a shock wave approaching a stationary state. *USSR Comp. Math. Math. Phys.*, 1:1187–1219, 1961.

[38] S. Gulati. *Effects of Abrupt Changes in Microfluidic Geometry on Complex Biological Fluid Flows*. PhD thesis, University of California, Berkeley, 2007.

[39] S. Gulati, S. J. Muller, and D. Liepmann. Direct measurements of viscoelastic flows in micro-contractions. In *Proceedings of the 3rd International Conference on Microchannels and Minichannels*, Toronto, Ontario, Canada, June 2005.

[40] S. Gulati, S. J. Muller, and D. Liepmann. Quantifying viscoelastic behavior of DNA-laden flows in microfluidic systems. In *Proceedings of the 3rd Annual International IEEE EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, Kahuku, Oahu, HI, May 2005.

[41] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:12:2182–2189, 1965.

[42] H. Johansen and P. Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comp. Phys.*, 147:60–85, 1998.

[43] H. S. Johansen. *Cartesian Grid Embedded Boundary Finite Difference Methods for Elliptic and Parabolic Partial Differential Equations on Irregular Domains.* PhD thesis, University of California, Berkeley, 1997.

[44] D. D. Joseph. *Fluid Dynamics of Viscoelastic Liquids.* Springer-Verlag, New York, 1990.

[45] D. D. Joseph, M. Renardy, and J.-C. Saut. Hyperbolicity and change of type in the flow of viscoelastic fluids. *Arch. Rat. Mech. Anal.*, 87:213–251, 1985.

[46] M. F. Lai. *A Projection Method for Reacting Flow in the Zero Mach Number Limit.* PhD thesis, University of California, Berkeley, 1994.

[47] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics.* Pergamon Press, Oxford, 1959.

[48] L. G. Leal. The motion of small particles in non-Newtonian fluids. *J. Non-Newtonian Fluid Mech.*, 5:33–78, 1979.

[49] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, New York, 2002.

[50] C. W. Macosko. *Rheology Principles, Measurements, and Applications.* VCH, New York, 1994.

[51] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Prentice-Hall, Englewood Cliffs, NJ, 1969.

[52] J. M. Marchal and M. J. Crochet. A new mixed finite element for calculating viscoelastic flow. *J. Non-Newtonian Fluid Mech.*, 24:85–96, 1987.

[53] D. F. Martin, P. Colella, and D. T. Graves. A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions. In preparation, 2007.

[54] P. McCorquodale, P. Colella, and H. Johansen. A Cartesian grid embedded boundary method for the heat equation on irregular domains. *J. Comp. Phys.*, 173:620–635, 2001.

[55] G. H. Miller. An iterative Riemann solver for systems of hyperbolic conservation laws, with application to hyperelastic solid mechanics. *J. Comp. Phys.*, 193:198–225, 2003.

[56] G. H. Miller and P. Colella. A high-order Eulerian Godunov method for elastic-plastic flow in solids. *J. Comp. Phys.*, 167:131–176, 2001.

[57] G. H. Miller and P. Colella. A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing. *J. Comp. Phys.*, 183:26–82, 2002.

[58] M. L. Minion. On the stability of Godunov-projection methods for incompressible flow. *J. Comp. Phys.*, 123:435–449, 1996.

[59] W. Noll. A mathematical theory of the mechanical behavior of continuous media. *Arch. Rat. Mech. Anal.*, 2:197–226, 1958.

[60] J. G. Oldroyd. On the formulation of rheological equations of state. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 200:523–541, 1950.

[61] R. G. Owens and T. N. Phillips. *Computational Rheology.* Imperial College Press, London, 2002.

[62] T. T. Perkins, D. E. Smith, and S. Chu. Single polymer dynamics in an elongational flow. *Science*, 176:2016–2021, 1997.

[63] N. Phan-Thien. *Understanding Viscoelasticity.* Springer, New York, 2002.

[64] N. Phan-Thien and R. I. Tanner. A new constitutive equation based derived from network theory. *J. Non-Newtonian Fluid Mech.*, 2:353–365, 1977.

[65] T. N. Phillips and A. J. Williams. Viscoelastic flow through a planar contraction using a semi-Lagrangian finite volume method. *J. Non-Newtonian Fluid Mech.*, 87:215–246, 1999.

[66] T. N. Phillips and A. J. Williams. Comparison of creeping and inertial flow of an Oldroyd B fluid through planar and axisymmetric contractions. *J. Non-Newtonian Fluid Mech.*, 108:25–47, 2002.

[67] J. J. Quirk. An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies. *Comput. Fluids*, 23:125–142, 1994.

[68] R. J. Rider. Filtering non-solenoidal modes in numerical solutions of incompressible flows. *IJNMF*, 28:789–814, 1998.

[69] R. S. Rivlin and J. L. Ericksen. Stress-deformation relations for isotropic materials. *J. Rat. Mech. Anal.*, 4:323–425, 1955.

[70] S. L. Rosen. *Fundamental Principles of Polymeric Materials.* John Wiley & Sons, New York, 1993.

[71] P. Schwartz, M. Barad, P. Colella, and T. Ligocki. A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions. *J. Comp. Phys*, 211:531–550, 2006.

[72] D. E. Smith, H. P. Babcock, and S. Chu. Single polymer dynamics in steady shear flow. *Science*, 283:1724–1727, 1999.

[73] K. L. Tan and C. Tiu. Entry flow behaviour of viscoelastic fluids in an annulus. *J. Non-Newtonian Fluid Mech.*

[74] R. I. Tanner and K. Walters. *Rheology: An Historical Perspective.*

[75] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, New York, 1999.

[76] M. Torrilhon. Exact solver for Riemann problems of ideal magnetohydrodynamics. Technical Report 2002-06, Seminar for Applied Mathematics, ETH, Zurich, Switzerland, April 2002.

[77] D. Trebotich, P. Colella, and G. H. Miller. A stable and convergence scheme for viscoelastic flow in contraction channels. *J. Comp. Phys.*, 205:315–342, 2005.

[78] E. H. Twizell, A. B. Gumel, and M. A. Arigu. Second-order, l_0-stable methods for the heat equation with time-dependent boundary conditions. *Advances in Computational Mathematics*, 6:333–352, 1996.

[79] J. S. Ultman and M. M. Denn. Anomalous heat transfer and a wave phenomenon in dilute polymer solutions. *Transactions of the Society of Rheology*, 14:307–317, 1970.

[80] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comp. Phys.*, 32:101–136, 1979.

[81] K. Weissenberg. A continuum theory of rheological phenomena. *Nature*, 159:310–311, 1947.

[82] J. L. White and A. B. Metzner. Development of constitutive equations for polymeric melts and solutions. *J. Appl. Polymer Sci.*, 7:1867–1889, 1963.

[83] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comp. Phys.*, 54:115–173, 1984.