A NUMERICAL METHOD FOR TWO-PHASE FLUID FLOWS WITH PARTICULAR APPLICATION TO PLANAR JETS

by

Allan Majdanac
B.Sc., Simon Fraser University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF MECHANICAL ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
August, 2004
©Allan Majdanac, 2004
In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Allan Majdanac
Name of Author (please print)

03/09/2004
Date (dd/mm/yyyy)

Title of Thesis: A Numerical Method for Two-Phase Fluid Flows with Particular Application to Planar Jets

Degree: MASC Year: 2004

Department of Mechanical Engineering
The University of British Columbia
Vancouver, BC Canada
Abstract

In modern paper making, the break-up of jets issuing from headboxes is responsible for a degradation in paper quality. Waves present on the jet surface are believed to initiate break-up. Numerical simulation of headbox flows and jets would allow prediction of surface waves and help control paper quality. The thesis objective is to significantly contribute to this task by adding two-phase flow capabilities into an existing finite-volume based fluid-flow solver. The solver is then used to numerically verify experimentally observed surface waves on planar water jets issuing from parallel channels.

The Volume of Fluid method is used to reformulate the equations of motion of both the liquid and gas phases into equations for a single composite fluid that describe the behavior of both individual phases simultaneously. The phases are identified within the composite fluid by an indicator field known as the volume fraction. The composite fluid retains the incompressibility and Newtonian behavior of its constituent fluids. The fluid interface is replaced by a discontinuity in the physical properties of the composite fluid, and the reformulation satisfies all boundary conditions associated with standard two-phase flow problems.

The reformulated equations are solved by standard finite-volume techniques, while the solution of the volume fraction equation requires special care. Several numerical two-phase techniques are presented and discussed. The CICSAM method is adopted for discretization of the volume fraction equation due to its impressive interface reconstruction abilities and its ease of compatibility with the existing code. The inclusion of surface tension is done through the Continuum Surface Force method, whereby a volumetric force mimicking surface tension effects is added to the momentum equations. These procedures are used in the existing solver and the behavior of the code is verified for a series of standard test cases. The solver exhibits very good quantitative and qualitative agreement with published results.

The modified code is used to simulate jets under various conditions. Jet thicknesses are small compared to the breadth to allow two-dimensional simulations to be conducted. Results for jets in the absence of gravity or surface tension are in excellent agreement with available analytical results. The inclusion of gravitational effects poses no additional con-
cerns. The addition of surface tension causes dramatic changes in jet interface profiles and increased difficulty in obtaining convergent results. A full two-dimensional simulation of water jets issuing from parallel channels with gravitational and surface tension effects is performed for domains of various downstream lengths. The experimentally observed surface waves cannot be reproduced by the tests performed in this study, due to an insufficient downstream grid length or possible three-dimensional effects. Suggestions for improvements relating to two-phase numerical routines and further simulations of planar jets are given.
Contents

Abstract

Table of Contents

List of Figures

List of Tables

List of Notation

Acknowledgments

1 Introduction 1
  1.1 The Paper Manufacturing Process ........................................... 1
  1.2 Instabilities in Headbox Jet Flows ........................................... 2
  1.3 Thesis Scope and Objectives .................................................. 5

2 Mathematical Formulation 7
  2.1 Two-Phase Fluid Flows .......................................................... 7
     2.1.1 Governing Equations .................................................... 7
     2.1.2 Boundary Conditions .................................................... 8
  2.2 The Volume of Fluid Method .................................................. 10
     2.2.1 Basic Assumptions .......................................................... 10
     2.2.2 Governing Equations .................................................... 11
     2.2.3 Surface Tension Effects .................................................. 14
     2.2.4 Boundary Conditions Revisited ........................................ 16
  2.3 Chapter Summary ................................................................. 18
# List of Figures

1.1 Modern headbox geometry. (a),(b): Tube bank, (c): Converging section. (From Söderberg [41]) ................................................. 2
1.2 Schematic of modern papermaking process (From Shariati [40]) ........ 3
1.3 Graphical description of streak sources. (a) Vortex stretching, (b) Centrifugal instability, (c) Boundary layer instability, (d) Wave instability. (From Söderberg [41]) .................................................. 4
1.4 Visualization of surface waves on planar jets. Waves appear as alternating light and dark horizontal bands. (From Söderberg [41]) ........................... 5

2.1 Representation of fluid interface showing height function $h(x, z, t)$ ........ 8
2.2 Representation of the transition region for surface tension volume force near interface (from Brackbill et. al. [3]) ............................... 15

3.1 General grid cell with pressure and curvilinear velocity components (from He [14]) .......................................................... 20
3.2 Cell naming convention for discretization procedures ....................... 22
3.3 Example of free surface flow profile computed by interface tracking (from Muzaferija and Peric [28]) ........................................ 24
3.4 Example of a result using an interface capturing method with a fixed grid (from Muzaferija and Peric [28]) .................................. 25
3.5 Examples of interface reconstructions using SLIC method (From Ubbink [49]) 27
3.6 Operator split dependence of SLIC method (From Rudman [38]) ....... 27
3.7 Interface reconstruction using Youngs’ method (from Rudman [38]) ........ 28
3.8 Interface reconstruction using piecewise linear method (From [10]) ....... 29
3.9 Comparison of a SLIC reconstruction with a PLIC method for a circle. Local volume fraction values are given and shaded. (From Kothe and Rider [37]) . 30
3.10 Computation of advected profiles ........................................... 32
LIST OF FIGURES

3.11 Allowable TVD regions. a) Shaded TVD area and discretization curves: Beam-Warming (2nd-up), Lax-Wendroff (2nd-down), Fromm. b) Shaded Sweby TVD area. Minmod c) Superbee d) MC (From Leveque [25]) .................................. 35
3.12 TVD limiters used in advecting various profiles ........................................ 36
3.13 a) Naming convention used for normalized variables b) Corresponding normalized values (From Leonard [23]) .................................................. 37
3.14 NVD for standard discretizations: (1U=1st up, 1D=1st Down, 2U=2nd Up, 2C=2nd Down, F= Fromm) (From Leonard [22]) ........................................ 37
3.15 NVD for Minmod (dashed) and Superbee limiters (solid) (From Leonard [23]) 38
3.16 Universal Limiter for Courant number-dependent flows (From Leonard [21]) 39
3.17 NVD for Hyper-C method (From Leonard [21]) ......................................... 40
3.18 Naming convention used for donor-acceptor schemes ................................ 41
3.19 Interface reconstruction using donor-acceptor method (From Ubbink [49]) .... 42
3.20 Interface profile calculation using original VOF method (From Rudman [38]) .. 43
3.21 NVD for multi-dimensional Universal Limiter (From Ubbink [49]) .............. 52
3.22 Smoothing of volume fraction profiles a) initial distribution b) Smoothed distribution (From Williams [51]) .................................................. 54
3.23 Filter for convolution based smoothing of volume fraction (From Peskin [35]) 55

4.1 Initial and computed profiles for translation tests ....................................... 61
4.2 Velocity field and initial conditions for solid body rotation test ................... 63
4.3 Initial and computed profiles for rotation tests ......................................... 63
4.4 Initial profile and velocity field for shear test ......................................... 65
4.5 Volume fraction profiles for shearing tests ............................................. 66
4.6 Volume fraction profiles for 1000 forward and reversed shearing steps using different methods (From Rudman [38]) ........................................ 67
4.7 Results of surface tension test case using CSF method .............................. 69
4.8 Initial distributions using different meshes, 30 x 30 resolution .................. 72
4.9 Relaxation of Square Drop ................................................................. 75
4.10 Rayleigh-Taylor instability at various times ........................................... 78
4.11 Grid used for sloshing tests ................................................................ 79
4.12 Results for air-water sloshing ............................................................... 80
4.13 Sloshing in tank at various times ........................................................... 81

5.1 Schematic of domain and boundary conditions used for jet test ................ 86
5.2 Mesh used for jet simulation ................................................................... 88
LIST OF FIGURES

5.3 Computed solution for $Re = 200$ ................................................................. 89
5.4 Computed and exact solutions for $Re = 200$ ................................................. 89
5.5 Comparison of solutions generated with three time level and Crank-Nicolson methods, $Re = 200$ ................................................................. 91
5.6 Comparison of solutions with different density ratios, $Re = 200$ .................. 92
5.7 Computed profile for jet with gravity, $Re = 200$ ........................................... 93
5.8 Comparison of computed $Re = 200$ jet profiles with and without gravity. Black=gravity, Red=no gravity ................................................................. 94
5.9 Simulation of planar jets with gravity at different Reynolds numbers ............ 95
5.10 Comparison of planar jets with and without gravitational effects. Black=gravity, Red=without gravity ................................................................. 96
5.11 Calculation of planar jets for increased spatial dimension showing comparison with analytical results (All dimensions in meters) ................................. 98
5.12 Planar jet with surface tension at various times, $Re=200$ ............................ 100
5.13 Mesh used for jet simulation in surface tension tests .................................... 102
5.14 Jet with surface tension, compared with Tillet profile, $Re = 1100$ ............. 102
5.15 Experimental setup for Soderberg's planar jet (From Söderberg [41]) ........... 105
5.16 Computed profile for experimental jet, $Re = 1430$ ................................... 106
5.17 $Re = 1430$ jet showing numerical disturbances at various times ............... 107
5.18 Computed profile for experimental jet, $Re = 1860$ .................................... 108
5.19 Transient growth for extended domain jet .................................................... 108
5.20 Comparison of $Re = 1860$ jet with surface tension and gravity to Tillet's analytical result, 5 diameter downstream length ........................................ 110
List of Tables

4.1 Comparison of error values for translation tests ....................... 61
4.2 Comparison of error values for rotation tests .......................... 64
4.3 Comparison of error values for shear tests ............................. 66
4.4 Error values for surface tension tests ................................. 71
4.5 Error values for surface tension tests using different laplacian filter passes ... 71
4.6 Error values for surface tension tests using circular meshes  .......... 72
4.7 Error values for surface tension tests with varying density ratios, rectangular mesh .................................................. 73
4.8 Error values for surface tension tests with varying density ratios, circular mesh 73

5.1 Dimensions and boundary conditions for jet simulations .............. 88
5.2 Error values for jet test, $Re = 200$ ......................................... 90
5.3 Error values for different density ratios, $Re = 200$ ..................... 91
5.4 Error values for higher Reynolds number jets with gravity ............ 96
5.5 Dimensions and boundary conditions used in surface tension jet tests ... 101
List of Notation

\(c, c_t: \) Courant number, total Courant number
\(f:\) Body force
\(F_{\sigma A}: \) Surface force per unit area due to surface tension
\(F_{\sigma V}: \) Volumetric surface tension force
\(Fr:\) Froude number
\(g:\) Gravitational acceleration
\(h:\) Channel width
\(H:\) Analytic free surface indicator function
\(n:\) Normal vector to fluid interface
\(\hat{n}:\) Unit normal vector to fluid interface
\(p:\) Pressure field
\(R:\) Radius of curvature
\(Re:\) Reynolds number
\(S:\) Rate of strain tensor
\(\mathbf{t}_{l,g,\sigma}^b:\) Normal stress vector of liquid, gas, surface tension
\(\mathbf{t}_{l,g}^t:\) Tangential stress vector of liquid, gas
\(\mathbf{T}_{l,g}:\) Stress tensor of liquid, gas phases
\(u:\) Velocity field
\(We:\) Weber number
\(x:\) Position vector
\(x':\) Interfacial position vector
\(\alpha:\) Volume fraction field
\(\beta_f: \) CICSAM weighting factors
\(\gamma_f: \) CICSAM blending constant
\(\Gamma: \) Interfacial transition region
\(\delta: \) Kronecker delta tensor
\(\delta(x - x'):\) Interfacial delta function
LIST OF NOTATION

\( \zeta \): Analytic jet interface profile
\( \theta \): TVD smoothness variable
\( \kappa \): Interface curvature
\( \mu_{l,g} \): Dynamic viscosity of liquid, gas
\( \rho_{l,g} \): Density of liquid, gas
\( \sigma \): Surface tension coefficient
\( \mathbf{\tau} \): Viscous stress tensor
\( \phi \): Numerical solution variable
\( \hat{\phi} \): Normalized variable
Acknowledgments

This thesis came to be through the assistance and support of many kind people. I would like to thank my supervisors, Dr. Ian Gartshore and Dr. Martha Salcudean, for their encouragement and support throughout the entire course of this study. Their guidance, confidence in my abilities, and words of wisdom have helped me immeasurably. Their patience with me as I ran off to do many other things not related to this thesis is also to be highly commended.

I would like to thank Dr. Paul Nowak for all of his help in guiding me through the ins and outs of his code. There were many times where I would have been completely lost without any assistance or information from him. His kindness is greatly appreciated. I would also like to extend thanks to Dr. Carl Ollivier-Gooch for teaching me a great deal about Computational Fluid Dynamics through his excellent courses, and also through all of the times that I constantly pestered him about one thing or another.

I would like to give a great deal of thanks to my family who have always believed in me and given me the opportunity to do the things in life I wanted to do. Their undying support and belief in me has helped me through a great deal, and I look forward to the fact that they will always continue to do so in the future as well. I will always love and support them, and I'll be lucky if I can help them as half much as they've helped me.

Finally, I would like to thank my loving girlfriend, Dragana, for being extremely patient with me through all of the bugs and problems that I've encountered while working on this thesis. Now that it's done, I'm guessing that I won't be continually late for everything because I'll no longer be setting a program to run or checking a result (but I'm sure I'll find other reasons....). Also, I hope to stop being an absentee boyfriend, seeing as how the "other woman" is now out of the picture :-)

xiii
Chapter 1

Introduction

1.1 The Paper Manufacturing Process

Paper manufacturing is a complex process whose fundamental principle is quite straightforward. The essential idea is that a suspension of water and some type of fibre is spread over a permeable surface (such as a fine screen). The water is permitted to pass through, while the fibres are trapped on the surface. Through a natural chemical process the trapped fibres will adhere to each other. Once the suspension has completely dewatered and dried, the remaining product is a thin sheet of paper. Though the idea is quite simple, the mechanical process by which the paper is formed has drastically varied throughout history.

Originally, paper sheets were hand formed, whereby a pulp suspension was manually spread over a permeable surface and drained from below through the action of gravity. This process is very time consuming and produces asymmetrical results due to the fact the drying of the paper occurs only on one side of the sheet. Today, paper sheets are formed through a continuous process aided by a series of several mechanical components. One of the more important of these components is the *headbox*. The headbox has a converging section that essentially acts as a nozzle with a thin, planar exit. A typical headbox geometry is shown in Figure 1.1. The pulp suspension is passed through the other components in the series and makes its way into the headbox through a number of small tubes known as a *tube bank*. Once the suspension enters from the tube bank, it is accelerated through the converging section and expelled from the headbox as a planar jet. Typical industrial headbox jets have exit widths of 5-10m and thicknesses of 1cm. The flow within the headbox itself is turbulent and anisotropic due to the distribution of the tube bank, giving jets with turbulent flow fields and velocities ranging from 10-35m/s. The resulting jet is expelled into a gap between two permeable surfaces, and the dewatering process takes place on both surfaces through
pressing and rolling of the sheet. The process is assisted by additional drying, and yields a more symmetrical final product. A schematic diagram of the process is given in Figure 1.2.

The quality of the final paper sheet will vary from method to method and depends greatly on the orientation of the fibres within the sheet. It is observed that semi-coherent streamwise orientations of the fibres exist in the flow of the jet and in the final paper sheet [41]. These are simply semi-organized patterns in the flow direction, known as streaks. The streaks create regions of non-uniformity in the sheet, lowering its overall quality. The fibre orientation, in turn, is determined by the nature of the flow over the surface as the fibres settle. For example, the level of turbulence within the flow field has a major effect on the flow field characteristics and the settling of the fibres. This means the formation and quality of the sheet are directly related to hydrodynamic considerations that arise from both the flow within the headbox and the planar jet itself. Determining the origin of these streaks is therefore a key issue in understanding the fluid mechanics involved in paper manufacturing and in improving the quality of the paper produced in the process.

1.2 Instabilities in Headbox Jet Flows

One can look at many flows in Newtonian fluid dynamics to observe streaks and vortices. Although the pulp suspension itself is a non-Newtonian fluid, it is nonetheless informative to investigate the conditions under which these streaks arise in Newtonian flows. This provides a solid understanding of the dynamical processes involved in the basic flowfield, and can help
clarify the physical behavior once non-Newtonian effects are to be included in the study of streak generation in paper manufacturing. Söderberg [41] provides an excellent summary of the potential origins of streaks and vortices in Newtonian fluids and their relation to headbox flows and planar jets. In brief, there are four possible sources of vortices and streaks in Newtonian flows, which are represented in Figure 1.3

- **Vortex Stretching**: This vortex generation mechanism is a result of the flow being accelerated within the converging section of the headbox and depends on the upstream conditions of the flow within the headbox (i.e., the tube bank flow imprint). It gives rise to streamwise oriented patterns in the flowfield.

- **Centrifugal instabilities**: These vortices arise due to streamline curvature within the headbox. A centrifugal force will act radially on the fluid, leading to an unstable stratification of the fluid that can outweigh the opposing effects of viscous retardation and radial pressure gradients. The imbalance of forces will lead to a subsequent onset of instabilities and streamwise vortices.

- **Boundary layer instabilities**: This type of instability arises within the boundary layer of the headbox walls. Small disturbances (due to wall roughness, etc.) within the boundary layer give rise to transient instabilities that lead to streaks oriented in the streamwise direction in the final flow.

- **Instability waves**: These waves occur in the exiting planar jet as its velocity profile relaxes further downstream. They are observed for both water jet flows and actual headbox flows.
CHAPTER 1. INTRODUCTION

Figure 1.3: Graphical description of streak sources. (a) Vortex stretching, (b) Centrifugal instability, (c) Boundary layer instability, (d) Wave instability. (From Söderberg [41])

Experimental work by Söderberg [41] shows that the expected order of magnitude (i.e., the boundary layer thickness) for streaks generated in centrifugal or boundary layer instabilities do not exist in headbox flows, and further concludes that these effects are not a cause of the final streaks. It is observed that some indications of vortex stretching effects due to upstream conditions generated by the tube bank are present. However, the evidence is not conclusive enough to assume that this is the sole cause of streaks. In fact, further experimental work by Shariati [40] shows that the imprint of the mean velocity wake of the tube banks is small, and decreases drastically before entering the headbox converging section. However, any non-uniformity and particularly the turbulence footprint in the flow pattern from the tube banks does tend to remain through the converging section, meaning there is still a possibility that the tube bank imprint could lead to streamwise streaks.

Due to stronger experimental evidence, Söderberg believes that the dominant source of the streaks are the instability waves in the jet, which are explained by linear stability theory. An example of the surface waves for various jets is shown in Figure 1.4. As the waves propagate downstream, they can grow in amplitude and lead to a break-up of the jet. This ultimately gives streamwise streak formation in the jet flow pattern and possibly in the final paper sheet itself. Furthermore, Söderberg finds that for low Reynolds number planar water jets, the break-up is localized in regions where the upstream flow inhomogeneity is greater, meaning that the surface waves appear to act as an amplifier to upstream disturbances from
CHAPTER 1. INTRODUCTION

a.  

b.  

c.  

d.  

Figure 1.4: Visualization of surface waves on planar jets. Waves appear as alternating light and dark horizontal bands. (From Söderberg [41])

the headbox. This implies that the jet is essential to streak formation, but it is also highly dependent on the flow conditions within the headbox. Nonetheless, there is still uncertainty as to the fundamental cause of the streaks themselves, whether or not the conditions in the headbox affect the final streak formation, the jet instabilities are the main source, or perhaps other unknown factors also contribute to their formation. It appears that in all cases the jet behavior plays a critical role.

1.3 Thesis Scope and Objectives

Surface waves on the headbox jet are experimentally observed [40, 41] and regardless of any debate concerning the ultimate origin of the streaks, it is apparent that the jet does have an important effect on the final sheet quality. While there have been experimental studies concerning headbox flows and the planar jet, there have not been any studies (to
the author's knowledge) that attempt to simulate the free planar jet from a headbox and numerically verify the presence of the observed surface waves. This provides the motivation for the current work. More specifically, the objectives are as follows:

1. To modify an existing CFD research code to include full two-phase flow capabilities

2. To simulate surface waves found on planar water jets exiting from channels, as observed by Söderberg [41]

The first item is the main objective and comprises the bulk of the thesis work and discussion. Along with the second objective, these are significant steps towards a long term goal of our research group; namely, to simulate the jets and surface waves that arise in industrial headboxes under actual operating conditions. The existing research code, CMGFD, employed by our group at the University of British Columbia, is a complex three-dimensional computer program that has been used to numerically investigate headbox flows [40]. The current work is a natural complement to this prior study, whereby we now attempt to investigate the predictability of planar jets and numerically verify their observed properties with regards to headbox flows. The project attempts to bring the work of the research group one step closer to simulating the behavior of the headbox flows, planar jets, and hence a large portion of the entire paper manufacturing process. The simulations, in turn, can provide valuable insight into the physical processes involved with the headbox and jet, and help make suggestions that can improve overall paper quality.
Chapter 2

Mathematical Formulation

2.1 Two-Phase Fluid Flows

2.1.1 Governing Equations

To model and simulate the flow phenomena related to planar jets, a mathematical description of the physical processes involved in two-phase flow is necessary. In order to simplify the analysis and computation, we assume that we are dealing with incompressible Newtonian fluids only. Since we are interested in investigating the dynamical behavior of this class of fluid, the governing equations for mass and momentum conservation are the incompressible continuity equation and the Navier-Stokes equations

\[ \nabla \cdot \mathbf{u} = 0 \]

\[ \rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \]

The above equations hold separately for each of the phases in the flow. The density, \( \rho \), and dynamic viscosity, \( \mu \), of each phase is assumed constant. Volumetric forces such as gravity are included in the final term of (2.2). As an additional simplification, a constant temperature is assumed throughout all phases and the effects of heat transfer within the flowfield are ignored. Furthermore, we study only laminar flow regimes, so there is no need to use additional mathematical models to describe the turbulence quantities within the jet. While industrial headbox and jet flows are turbulent, we are interested in simulating surface waves for low Reynolds number (laminar) cases that can be compared with suitable experimental results. This simplification is used in an attempt to understand the fundamental mechanism involved in the generation of these waves without the additional complications of a turbulent flowfield.
2.1.2 Boundary Conditions

In a two-phase flow regime, the different phases are separated by a well defined boundary or interface. This is essentially a discontinuity in the physical properties of the different fluids on a macroscopic level. Each phase of the flow is governed by (2.1) & (2.2), and the interface between them must obey certain boundary conditions. The standard analytical method of describing the location of the free surface is to prescribe a height function, \( H(x, y, z, t) = y - h(x, z, t) \), to the interface relative to a set of coordinate axes. The function \( h(x, z, t) \) is unknown, and the total function \( H(x, y, z, t) \) locates the interface in space and time. A sketch of the interface and height function is given in Figure 2.1.

![Figure 2.1: Representation of fluid interface showing height function \( h(x, z, t) \)](image)

As mentioned, the height function is unknown and must be solved for in addition to the flowfield velocity and pressure. To do this, boundary conditions at the interface are required. The height function is governed by the kinematic boundary condition

\[
\frac{DH}{Dt} = \frac{\partial H}{\partial t} + (u \cdot \nabla)H = 0
\]

This condition states that the total rate of change (the material derivative) of \( H(x, y, z, t) \) is zero and that any particle on the interface must move with the velocity of the interface.

As an additional boundary condition, there must be a balance of forces on the interface itself. This is the dynamic boundary condition, which states that the difference in normal
stresses between the liquid and gaseous phase must be balanced by the stress due to surface tension

\[ t_i^n - t_g^n = t^s_n \] (2.4)

The stresses of the liquid and gaseous phases are given by projecting the corresponding stress tensor, \( \mathbf{T} \), onto the unit vector normal to the interface, \( \mathbf{n} \)

\[ t^n_{i,g} = \mathbf{T}_{i,g} \cdot \mathbf{n} \] (2.5)

The stress tensor of an incompressible fluid is the sum of the hydrostatic pressure and viscous stress tensor

\[ \mathbf{T} = -p\delta + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \] (2.6)

where \( \delta \) is the Kronecker delta tensor. The unit vector normal to the interface, \( \mathbf{n} \), is found by normalizing the gradient of the surface location

\[ \mathbf{n} = \frac{\nabla H}{|\nabla H|} \] (2.7)

In general, the stress due to surface tension is a complicated function of the surface tension coefficient, \( \sigma \), having components in the normal and tangential directions to the interface [3, 9]. If we consider the coefficient to be constant, then the stress due to surface tension reduces to a term in the normal direction that is represented as

\[ t^s_n = \sigma \kappa \mathbf{n} \] (2.8)

where \( \kappa \) is the curvature of the interface, given by

\[ \kappa = -\nabla \cdot \mathbf{n} \] (2.9)

Furthermore, the dynamic boundary condition states that the tangential components of stress across the interface must be equal for a constant surface tension coefficient

\[ t_i^t = t_g^t \] (2.10)

These boundary conditions, coupled with the mass and momentum conservation equations constitute a complete description of the two-phase flow problem for a jet. However, there are difficulties in the solution of this system of equations. The surface location is unknown and must be solved for simultaneously with the other equations. The boundary conditions
(2.3), (2.4), (2.10) are all non-linear, which complicates the solution process tremendously. There are no general analytic solutions and a numerical solution is quite difficult, as (2.1), (2.2) need to be solved for both phases in addition to satisfying the boundary conditions (2.3), (2.4), (2.10).

2.2 The Volume of Fluid Method

2.2.1 Basic Assumptions

An alternate method for describing the physical problem of two-phase flow is taken from a class of numerical free surface methods known as interface-capturing methods [9, 28]. The idea is to avoid the complication of using two separate phases and an explicit boundary by redefining the system to contain only a single composite fluid with discontinuous physical properties. The nature of the discontinuity describes the interface dividing the two fluids. This is known as the Volume of Fluid (VOF) method, and was originally proposed by Hirt and Nichols [15] for the numerical solution of two-phase flow problems. It has become a standard method for dealing with multi-phase flow simulations, and is described in [10, 15].

The interface is defined by an indicator function, known as the volume fraction, $\alpha$. It represents the ratio of the volume of liquid in a fluid element, $\Delta V_i$, to the volume of the element itself, $\Delta V$. Regions consisting solely of liquid will have a volume fraction of unity, while regions of gas will have a volume fraction of zero. The interface is located in the transition regions where $0 < \alpha < 1$. Any values below zero and above one are non-physical, and are to be avoided. The volume fraction is therefore a scalar field in space and time, $\alpha(x, t)$, which defines the distribution of the two fluids. The indicator field is propagated with the fluid itself and is considered to be a Lagrangian invariant [15], meaning that its material derivative is zero

$$\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla) \alpha = \frac{D\alpha}{Dt} = 0 \quad (2.11)$$

Equation (2.11) is the advective form of the governing equation for the indicator field in this composite fluid method. The volume fraction formulation describing multi-fluid problems can easily be extended to include more than two phases by defining volume fractions for each additional phase beyond single phase flow and a corresponding equation, analogous to (2.11) [10]. Furthermore, one must require that the total sum of all volume fractions in a local fluid element be unity. The present study will only consider the case of two separate fluids.

With the different phases specified by the volume fraction distribution, and the behavior
of the indicator field determined through (2.11), we define the physical properties of the composite fluid to match those of the two explicit fluids. We have a single fluid that has discontinuous density and viscosity in accordance with the distribution of the two fluids. The physical parameters are represented as arithmetic averages of the properties of each phase

\[ \rho = \rho_1 \alpha + \rho_2 (1 - \alpha) \]  
\[ \mu = \mu_1 \alpha + \mu_2 (1 - \alpha) \]

The representation of viscosity by an arithmetic average is not strictly correct as stress continuity is not enforced. A harmonic average of both viscosities is the correct approach. However, in practice there is little difference between the two averages, and the simpler arithmetic average is used to represent the viscosity in this study.

### 2.2.2 Governing Equations

The constituents of the composite fluid are both incompressible and Newtonian. However, the global density and viscosity are no longer constant, so the equations of mass and momentum conservation must be re-examined. The general form of mass conservation is a scalar conservation law for density that (in the absence of mass sources) states

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \]

The last term on the left hand side of (2.14) is expanded by noting that

\[ \nabla \cdot (\rho \mathbf{u}) = (\mathbf{u} \cdot \nabla) \rho + \rho (\nabla \cdot \mathbf{u}) \]

Substituting (2.15) into (2.14) and grouping terms to form a material derivative for the density [49], the modified form of (2.14) is written as

\[ (\nabla \cdot \mathbf{u}) = -\frac{1}{\rho} \frac{D\rho}{Dt} \]  
\[ (\nabla \cdot \mathbf{u}) = -\frac{\rho_1 - \rho_2}{\rho} \frac{D\alpha}{Dt} = 0 \]

The composite density (2.12) is substituted into (2.16) to give the following
This result is obtained by noting that the material derivative of the volume fraction is zero by (2.11). It follows that the composite fluid is also an incompressible fluid, regardless of the fact that it has discontinuous physical properties. Hence, mass conservation for the composite fluid is represented by (2.1) as opposed to the full continuity equation (2.14). This is a very useful simplification inherent to the Volume of Fluid method.

Since the fluid is incompressible, we add a factor of \( \alpha (\nabla \cdot \mathbf{u}) \) to (2.11) without changing its behavior:

\[
\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla) \alpha + \alpha \nabla \cdot \mathbf{u} = \frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \tag{2.18}
\]

This is known as the conservative form of the volume fraction equation. The evolution of the volume fraction field is therefore governed by a scalar conservation law, just as with the density in (2.14). The conservative form of the volume fraction equation (2.18) is mathematically equivalent to (2.11), but is more suitable to finite volume numerical discretization than the advective form.

The general form of momentum conservation for a fluid with variable density and viscosity is given by Navier’s equation \([50]\) for a viscous fluid:

\[
\frac{D(\rho \mathbf{u})}{Dt} = -\nabla p + \mathbf{\nabla} \cdot \mathbf{\tau} + \rho \mathbf{f} \tag{2.19}
\]

The term \( \mathbf{\tau} \) represents the viscous stress tensor, which is discussed below. Expansion of the material derivative in (2.19) gives

\[
\frac{D(\rho \mathbf{u})}{Dt} = \mathbf{u} \frac{\partial \rho}{\partial t} + \rho \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)(\rho \mathbf{u}) \tag{2.20}
\]

The convective term in (2.20) is expanded as

\[
(\mathbf{u} \cdot \nabla)(\rho \mathbf{u}) = \mathbf{u} (\mathbf{u} \cdot \nabla \rho) + \rho (\mathbf{u} \cdot \nabla \mathbf{u}) \tag{2.21}
\]

Substitution of (2.21) and (2.20) into the left hand side of (2.19) yields

\[
\rho (\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla \mathbf{u})) + \mathbf{u} (\frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla \rho)) \tag{2.22}
\]

The second term in (2.22) vanishes due to mass continuity, (2.14), and the incompressibility of the fluid, (2.1). Hence, the left hand side of Navier’s equation, (2.19), is simply a product of the density with the material derivative of the velocity, given by the first term in (2.22). The appropriate density for use within the VOF method is the discontinuous form, (2.12).
CHAPTER 2. MATHEMATICAL FORMULATION

The viscous stress tensor, $\tau$ has the following general form [50]

$$\tau = 2\mu \dot{S} + \zeta \delta \nabla \cdot u \quad (2.23)$$

where $\mu$ represents the dynamic molecular viscosity, $\dot{S}$ is the rate of strain tensor, $\zeta$ is the second viscosity, and $\delta$ is the Kronecker delta tensor. Since the fluid is incompressible, the term containing the second viscosity vanishes, and we obtain the expression that the viscous stress tensor is proportional to the rate of strain tensor, i.e., the composite fluid is Newtonian

$$\tau = 2\mu \dot{S} \quad (2.24)$$

We substitute this expression for the viscous stress tensor into (2.19). The viscosity, $\mu$, is the composite form, (2.13), so it is acted upon by the divergence operator

$$\rho \left( \frac{\partial u}{\partial t} + (u \cdot \nabla u) \right) = - \nabla p + \nabla \cdot (2\mu \dot{S}) + \rho f \quad (2.25)$$

The rate of strain tensor, $\dot{S}$, has the standard form

$$\dot{S} = \frac{1}{2} (\nabla u + (\nabla u)^T) \quad (2.26)$$

As the rate of strain tensor is symmetric, the divergence term of (2.25) yields the following quantities

$$\nabla \cdot (2\mu \dot{S}) = \nabla \mu \cdot \nabla u + \mu \nabla^2 u + \nabla \mu \cdot (\nabla u)^T + \mu \nabla (\nabla \cdot u) \quad (2.27)$$

Since the flow is incompressible, (2.27) is written as

$$\nabla \cdot (2\mu \dot{S}) = \nabla \mu \cdot (\nabla u + (\nabla u)^T) + \mu \nabla^2 u \quad (2.28)$$

Using the composite expression for viscosity, (2.13), the first right hand side term in (2.28) is rewritten as

$$\nabla \mu \cdot (\nabla u + (\nabla u)^T) = (\mu_t - \mu_g) \nabla \alpha \cdot (\nabla u + (\nabla u)^T) \quad (2.29)$$

Substitution of (2.29) into (2.25) yields the final form of the momentum equations in the Volume of Fluid method

$$\rho \frac{Du}{Dt} = - \nabla p + \mu \nabla^2 u + (\mu_t - \mu_g) \nabla \alpha \cdot (\nabla u + (\nabla u)^T) + \rho f \quad (2.30)$$
CHAPTER 2. MATHEMATICAL FORMULATION

If the viscosity is constant, the momentum equations (2.30) reduce to the standard Navier-Stokes form of (2.2), as expected.

A more useful form of the momentum equations (2.30) with respect to finite volume numerical discretization is the conservative form. To re-express (2.30) into this form, we begin by adding factors of $u \left( \frac{\partial \rho}{\partial t} + (u \cdot \nabla) \rho \right)$ and $\rho (\nabla \cdot u)$ to the left hand side of (2.25) and substitute in (2.26) to obtain

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot \left( \rho uu + p \delta - \mu \left( \nabla u + (\nabla u)^T \right) \right) = \rho f$$

(2.31)

2.2.3 Surface Tension Effects

We have reformulated an explicit two-phase flow problem into one involving a single composite fluid. In doing so, we have completely removed the actual interface between the fluids and replaced it with a discontinuity in physical properties which describes the original fluid-fluid interface. This method has many advantages over the explicit two-phase method. For example, in the height function formulation of two-phase flows, (2.3), it is not possible to use multi-valued functions, $h(x, z, t)$, (i.e. relations) to describe interface location. Hence, describing curves such as circles or flows where a fluid folds over itself is not possible with the method. Furthermore, the reduction in computational effort with the Volume of Fluid method is substantial, as there is only one set of equations to be solved in any given time step, as opposed to solving two sets of equations for an explicit two-phase flow system.

However, with the loss of the explicit boundary, a problem arises: namely, how does one incorporate surface tension effects into a fluid with no interface? This difficulty is overcome by reformulating these effects into a volumetric force that is added to the momentum equations as any other body force term, such as gravity. This allows us to use the Volume of Fluid method and benefit from all of its advantages, while still having accurate predictive capabilities with regards to interface dynamics.

The concept was originally proposed by Brackbill et. al. in the Continuum Surface Force (CSF) method [3]. The idea behind the method is to model the discontinuous interface between the fluids (represented by a step profile in the volume fraction) by a small, smoothly varying transition region of thickness $h$ (in which the volume fraction changes smoothly as well). In this model, it is no longer appropriate to limit the surface tension effects to a boundary condition applied only at the interface, but instead to localize them into a volumetric force, $F_{\sigma v}$, located within the transition region. In the limit as $h \rightarrow 0$, the volume force returns the correct interfacial stress due to surface tension, found from integrating the
surface force (force per unit area), \( F_{\sigma_A} \), over the surface area of the interface itself

\[
\lim_{{h \to 0}} \int \int \int_{\Delta V} F_{\sigma_v} dV = \int \int F_{\sigma_A} dA
\]  

(2.32)

\( F_{CT} \) is restricted to lie in \( \Gamma \) by a delta function of finite width, centered about the interface location

\[
F_{\sigma_v} = \int \int \int_{\Gamma} F_{\sigma_A} \delta(x - x') dV
\]  

(2.34)

Figure 2.2: Representation of the transition region for surface tension volume force near interface (from Brackbill et. al. [3])

The CSF method is visually represented in Figure 2.2. The volume force acting on the transition region, \( \Gamma \), (i.e. essentially the interface itself) is found by integrating the surface force, \( F_{\sigma_A} \), over the volume of the transition region. For a constant surface tension coefficient, \( \sigma \), the surface tension effects reduce to a force normal to the interface

\[
F_{\sigma_A} = \sigma \kappa \hat{n}
\]  

(2.33)
CHAPTER 2. MATHEMATICAL FORMULATION

The force has a non-zero value only inside the transition region due to the delta function, and is set to zero for each of the uniform fluids outside of the transition region. The delta function plays a large role in determining the value of $F_{av}$ and its use is discussed in greater detail in Chapter 3.

A disadvantage of having a sharp step profile in the volume fraction to represent the interface is that the evaluation of the interface curvature, $\kappa$, cannot be accurately done. The curvature is the divergence of the unit normal vector to the interface

$$\kappa = -\nabla \cdot \hat{n}$$

(2.35)

The unit normal is given by the gradient of the volume fraction field

$$\hat{n} = \frac{\nabla \alpha}{|\nabla \alpha|}$$

(2.36)

Hence, the volume fraction field must be twice differentiable in order to correctly model the curvature and hence the surface tension force, $F_{\sigma_A}$. Clearly, a step profile is not suitable for this task, as it changes its value over an infinitesimal distance. The remedy is to define a smoothed volume fraction field, $\tilde{\alpha}$, that gradually changes from zero to unity within the transition region, permitting accurate computation of the curvature and interface normal. As alluded to earlier, the smoothing of the interface is completely consistent with the idea of a smooth finite transition region representing the interface, and in fact, is the basis of the CSF method. The smoothed values are used in (2.33) to calculate the surface tension force, $F_{\sigma_A}$. The function used to perform the smoothing is critical to the computation of the quantities in (2.35), (2.36) and is discussed in detail in Chapter 3.

The momentum equations include a surface tension contribution (2.34) within the framework of the composite fluid method and have the resulting form

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho uu + p \delta - \mu (\nabla u + (\nabla u)^T)) = \rho f + F_{av}$$

(2.37)

2.2.4 Boundary Conditions Revisited

When using the Volume of Fluid method to describe our system, there is no longer an explicit boundary between the fluids. Hence, we have no opportunity to apply the necessary boundary conditions at the fluid-fluid interface. However, examination of VOF reveals that the conditions are inherently satisfied with this method.

To begin with, the volume fraction equation (2.11) is simply a re-expression of the kine-
matic boundary condition (2.3). The volume fraction serves as the function that denotes the location of the free surface in space and time, much like the height function assigned to the explicit two-phase case. Hence, the volume fraction and (2.11) are alternative, but equivalent, expressions for the height function and the kinematic boundary condition.

The dynamic boundary condition is recovered by examining the momentum equations (2.38). We use a reference frame attached to the interface [47], and rewrite (2.37) involving a material (Lagrangian) derivative and the surface tension volume force

$$\frac{D(\rho u)}{Dt} + \nabla \cdot \left( p\delta - \mu (\nabla u + (\nabla u)^T) \right) = \rho f + F_{\sigma V} \quad (2.38)$$

Integrating (2.38) over a small control volume $\Delta V$ [2]

$$\int \int \int_{\Delta V} \left[ \frac{D(\rho u)}{Dt} + \nabla \cdot \left( p\delta - \mu (\nabla u + (\nabla u)^T) \right) \right] dV = \rho f + F_{\sigma V} \quad (2.39)$$

By the Divergence theorem, the pressure and viscous terms are converted to surface integrals

$$\int \int \int_{\Delta V} \frac{D(\rho u)}{Dt} dV + \int \int \left( p\delta - \mu (\nabla u + (\nabla u)^T) \right) \cdot \text{n} dA = \int \int \int_{\Delta V} (\rho f + F_{\sigma V}) dV \quad (2.40)$$

We assume that the control volume has a cylindrical shape (a “pillbox”) with its top and bottom faces parallel to the interface. Letting the height of the control volume approach zero, its volume vanishes, and the Lagrangian derivative and body force terms approach zero. By definition of the surface tension force term, (2.32), for a vanishing control volume thickness, the volume force term, $F_{\sigma V}$, gives the correct interfacial force per unit area. Hence, only the surface tension term and surface area integral terms remain.

$$\int \int \left( p\delta - \mu (\nabla u + (\nabla u)^T) \right) \cdot \text{n} dA = \int \int F_{\sigma A} dA \quad (2.41)$$

We examine the normal components of the expression, (2.41). By (2.33), $F_{\sigma A} = \sigma \kappa \text{n}$, hence, evaluating the surface integrals for the normal components gives

$$\left[ p - \mu (\nabla u + (\nabla u)^T) \right]_l - \left[ p - \mu (\nabla u + (\nabla u)^T) \right]_g = -\sigma \kappa \quad (2.42)$$
This is precisely the normal component of the dynamic boundary condition (2.4). Similarly, examining the tangential components gives

\[
[p - \mu(\nabla u + (\nabla u)^T)]_i = [p - \mu(\nabla u + (\nabla u)^T)]_g \tag{2.43}
\]

(2.43) clearly states the tangential stresses across the interface are continuous, as in (2.10). Therefore, the VOF method does indeed satisfy the boundary conditions for a two-phase explicit flow problem (2.3), (2.4), (2.10), albeit in an implicit manner.

2.3 Chapter Summary

The Volume of Fluid method provides a useful simplification in describing two fluid systems by reducing the number of phases for which the governing equations are solved. The composite fluid retains the incompressibility and Newtonian stress-strain relationship of its component fluids. The behavior of the different phases is described by the volume fraction field and the equations of motion are reformulated to incorporate the modified definitions of the density and viscosity of the fluid, as based on the volume fraction

\[
\rho = \rho_i \alpha + \rho_g (1 - \alpha) \tag{2.44}
\]

\[
\mu = \mu_i \alpha + \mu_g (1 - \alpha) \tag{2.45}
\]

\[
\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla) \alpha = 0 \tag{2.46}
\]

\[
\nabla \cdot \mathbf{u} = 0 \tag{2.47}
\]

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot \left(\rho \mathbf{u} \mathbf{u} + p \delta - \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\right) = \rho \mathbf{f} + \mathbf{F}_{\sigma_v} \tag{2.48}
\]

Surface tension effects are easily incorporated into the framework of the VOF method by use of the CSF method. The boundary conditions for an explicit two-phase flow system are implicitly satisfied. Therefore, the Volume of Fluid method is a simple, useful, and powerful means of describing two-phase flow systems.
Chapter 3

Numerical Methods

Using the VOF method, we have a suitable and affordable (from the point of view of numerical computation) means of computing two-phase flow problems. However, there are several issues to address when implementing the necessary modifications into an existing flow field solver. The details of the approach taken are now discussed at length.

3.1 Numerical Flow Field Solver

The flow field solver used in this numerical study is a modification of the CMGFD code developed by Dr. Paul Nowak at the University of British Columbia [13, 30]. This code is used in previous studies of headbox flow [40], and the current multi-phase generalization is a natural step in its evolution and in the research work of our group. CMGFD is a three-dimensional, curvilinear coordinate based, fully coupled incompressible finite-volume flow solver. It uses structured meshes with domain segmentation and non-orthogonal grid techniques. CMGFD is capable of solving problems with constant or variable densities, heat transfer effects, and can also simulate the motion of turbulent flows through the use of the $k$-$\epsilon$ model and other blended two-equation turbulence models. As this study only deals with laminar, constant temperature flows, the implementation of heat transfer effects and turbulence models will not be discussed. Further details are found in [13, 14, 30].

The numerical grids used for solution are based on a general curvilinear coordinate system. The grids have a staggered formulation (i.e., a MAC scheme) [34], whereby scalar values (such as pressure and volume fraction) are stored at each cell center, while the curvilinear velocity components are given at each cell face. The cells themselves need not be regular or orthogonal for this storage method. A general cell is shown in Figure 3.1, along with the data storage points.
CHAPTER 3. NUMERICAL METHODS

Figure 3.1: General grid cell with pressure and curvilinear velocity components (from He [14])

The flow field is solved for using a coordinate invariant, conservative form of the governing equations of mass and momentum, as given in Chapter 2

\[ \nabla \cdot \mathbf{u} = 0 \quad (3.1) \]

\[ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + p \delta - \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) = \rho f + \mathbf{F}_\sigma \quad (3.2) \]

In order to apply the finite volume discretization method, (3.1), (3.2) are integrated over a small control volume, \( \Delta V \)

\[ \iiint_{\Delta V} (\nabla \cdot \mathbf{u})dV = 0 \quad (3.3) \]

\[ \iiint_{\Delta V} \left[ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + p \delta - \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) \right]dV = \iiint_{\Delta V} (\rho f + \mathbf{F}_\sigma)dV \quad (3.4) \]

Using Gauss’s theorem, the volume integrals of divergence terms are converted to surface integrals

\[ \oint \mathbf{u} \cdot ndA = 0 \quad (3.5) \]

\[ \iiint_{\Delta V} \frac{\partial (\rho \mathbf{u})}{\partial t}dV + \oint_{\Delta A} [\rho \mathbf{u} \mathbf{u} + p \delta - \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] \cdot ndA = \iiint_{\Delta V} (\rho f + \mathbf{F}_\sigma)dV \quad (3.6) \]
CHAPTER 3. NUMERICAL METHODS

Working with control volume averages, we replace the volume integral terms in (3.6), and directly solve for the average values of the body force and time derivative terms

$$\Delta V \frac{\partial (\rho u)}{\partial t} + \oint \left[ \rho \mathbf{u} \cdot \mathbf{n} - \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \right] \cdot \mathbf{n} dA = \Delta V (\rho \mathbf{f} + \mathbf{F}_s)$$

(3.7)

Here, \( \mathbf{n} \) is the normal vector to a given cell face. All continuous terms in (3.5) and (3.7), such as the velocity and pressure are replaced with their respective control volume averages, and these averaged values are solved for in the discretized formulation of the mass and momentum equations.

The surface integrals are essentially summations of the given quantities evaluated at a cell face, \( f \), summed over all faces in a given control volume. Hence, the equations become

$$\sum_f (\mathbf{u}_f \cdot \mathbf{n}_f) A_f = 0$$

(3.8)

$$\Delta V \frac{\partial (\rho u)}{\partial t} + \sum_f [\rho_f \mathbf{u}_f \mathbf{u}_f + p_f \mathbf{f}_f - \mu_f (\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T)] \cdot \mathbf{n}_f A_f = \Delta V (\rho \mathbf{f} + \mathbf{F}_s)$$

(3.9)

The momentum equations are discretized forward in time using an implicit first-order Euler method

$$\Delta V \frac{(\rho u)^{n+1}_p - (\rho u)^n_p}{\Delta t} + \sum_f [\rho_f \mathbf{u}_f \mathbf{u}_f + p_f \mathbf{f}_f - \mu_f (\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T)]^{n+1} \cdot \mathbf{n}_f A_f = \Delta V (\rho \mathbf{f} + \mathbf{F}_s)^{n+1}_p$$

(3.10)

The superscripts refer to the time level at which a given term is evaluated (\( n \) is the previous time step, while \( n + 1 \) is the current level). Note that with the VOF method, the grid is fixed in space and time (Eulerian), so there is no change in geometrical quantities such as cell volume, face areas, and cell face normal vectors.

When using the finite volume method to discretize equations, only the average value of a solution, \( \phi \), is known within a given cell. The method requires the solution values at the control volume faces, \( \phi_f \), as in (3.8) and (3.10), so there must be an interpolation of the solution there by an appropriate method. The spatial interpolations used in the CMGFD code are first-order upwinding, second-order central differencing, second-order upwind or the mixed positive (Minmod) scheme.

Solution interpolation by first order upwinding simply represents the face value as the upstream cell value

$$\phi_f = \phi_p$$

(3.11)
Second order central differencing is the most "natural" of second order methods. It assumes the face value can be represented as a linear interpolation of the two cell values straddling the face, $\phi_P$ & $\phi_E$. The method then approximates the face value with a downwind weighted slope [22] based on the naming convention of Figure 3.2

$$\phi_f = \phi_P + \frac{\phi_E - \phi_P}{d_{PE}} d_{pf}$$

(3.12)

Second order upwinding assumes a directional bias to the upwind side of the flow. Here, a linear interpolation between the cells $\phi_W$ & $\phi_P$ is used, giving an upwind weighted slope

$$\phi_f = \phi_P + \frac{\phi_P - \phi_W}{d_{WP}} d_{pf}$$

(3.13)

The Minmod scheme is a combination of second order upwinding and central differencing [25]

$$\phi_f = \phi_P + \sigma d_{pf}$$

(3.14)

$$\sigma = \max \left( \min \left( \frac{\phi_E - \phi_P}{d_{PE}}, \frac{\phi_P - \phi_W}{d_{WP}} \right), 0 \right)$$

This method compares the upwind and downwind weighted slopes at a given point and chooses the smaller of the two, so long as they are positive. This method is discussed further in Section 3.3. The second-order accurate methods (in particular second-order upwinding), are used throughout the course of this study.
The flow field solution is given in a primitive variable form and obtained using a block-structured implicit formulation based on the work of Vanka [43]. The pressure and velocity are coupled and solved for simultaneously using a line Gauss-Seidel iterative method. Further details regarding the solution method of the governing equations, grid generation, and curvilinear discretization procedures within the CMGFD program can be found in He et. al. [13] and He [14]. This flow solver has been successfully used within our research group for many years to solve a variety of research and industrial projects [14, 40], and its results have proved useful and been generally well accepted.

3.2 Interface Methods

The research code, CMGFD, is a single phase flow solver. Therefore, it needs to be modified in order to study two-phase flow problems. Several methods exist that are appropriate for locating an interface on a numerical grid. These methods can roughly be divided into two classes, interface capturing and interface tracking [9, 28].

In an interface tracking method, there are deformable control volumes which explicitly define the boundary of the fluid phase being solved for. To model two-phase flows, the governing equations (2.1), (2.2), are solved for both phases. The boundary conditions (2.3), (2.4), (2.10) are numerically applied to the interface in order to provide a coupling of the two phases. Interfaces computed by this method are extremely sharp and well defined, as the cell boundaries define the interface itself. An example of a deformable grid used in an interface capturing method is given in Figure 3.3, where an open channel, free surface flow profile over a semi-circular obstacle is computed. The inlet flow is subcritical and due to the constriction caused by the obstacle, the flow becomes supercritical, is accelerated and contracts.

As mentioned, interface tracking is a costly procedure, as two separate sets of calculations are performed in order to track the motion of both phases and locate the interface. Hence, this class of method is not implemented into the existing code, as the resulting solutions would be too computationally expensive and too time consuming to compute. Furthermore, the implementation of deformable grids into the existing framework of the CMGFD code would be very difficult and would not take advantage of the existing grid generation methods within the program.

Instead, the interface capturing class is used. A non-deforming, stationary grid is used to “capture” the interface by means of an indicator field. The indicator field used in this study is the volume fraction (as discussed in Section 2.2), and it describes the distribution
CHAPTER 3. NUMERICAL METHODS

Figure 3.3: Example of free surface flow profile computed by interface tracking (from Muzaferija and Peric [28])

of the two fluids in space and time by representing the amount of a given fluid within a grid cell by a value between zero and unity. When the volume fraction is used, the two fluids can be combined into a single composite fluid with discontinuous physical properties, given by (2.12), (2.13). Therefore, the interface capturing class (i.e., the VOF method) removes the explicit interface itself, as there is only one fluid phase to solve for and hence no boundary for which to apply any conditions. The standard two-phase flow boundary conditions are implicitly satisfied by the method, as shown in Section 2.4. In order to capture the motion of both phases, we solve the governing equations (2.1), (2.31) throughout the whole domain for the single composite fluid. This drastically reduces the computational effort needed to solve two-phase flow problems.

The volume fraction field is governed by the scalar transport equation, (2.18). This equation and its discretization are simple to implement and solve for within the framework of CMGFD, as the code uses scalar conservation equations for calculation of turbulence quantities and heat transfer effects. Hence, it is a logical choice to adopt the interface capturing class of method, as the techniques for solving the types of equations required are already in place within the code.

While the interface capturing class and VOF method are simple to use and solve for in principle, their major drawback is that, depending on the discretization method used for the volume fraction equation, the interface can only be located to an accuracy of within one to three grid cells at best [28, 48]. The interface can be smeared over many cells, potentially losing the discontinuous nature of the free surface itself. An example of a solution found
with an interface capturing (VOF) method is given in Figure 3.4, showing the counterpart to the interface tracking grid used in Figure 3.3. Note that while the solution has essentially the same profile as that of Figure 3.3, the interface is located over more than one cell.

According to Muzaferija and Peric [28], the accuracy of solution in the two different classes is comparable in cases where they are both applicable. However, the interface capturing method has far more utility in dealing with many classes of problem than the interface tracking method (for example, in flows that fold over themselves). While the smearing of the interface with capturing methods can be a problem, there are several techniques that attempt to keep a tight interface resolution (within one cell) while still benefiting from the convenient features of these methods. Most, if not all, modern approaches to two-phase flow simulation in either research or commercial CFD codes [10, 37, 38, 48, 49] use the VOF model or some type of similar interface capturing method to describe the location of the fluids in space and time.

3.3 Interface Reconstruction

When using an indicator field to define a fluid distribution within the framework of an interface capturing method (such as the volume fraction within the VOF method), the description of the interface is stored discretely through the volume fraction values. Only information about the amount of each fluid in a given cell remains and all knowledge of
the actual interface shape and orientation is lost. Hence, there must be a reconstruction of the interface using the volume fraction values to recover any knowledge concerning the original shape. This is the key issue in the development of interface capturing methods. As a result of extensive research, solution and reconstruction methods can be roughly grouped into three categories [49]: line methods, high-resolution discretization schemes, and donor-acceptor formulations. It is important to note that each of these methods have techniques which only apply in computational cells where the interface resides, i.e., where the volume fraction is between zero and unity (but use information from the nearest neighbor cells). For cells completely filled with a single fluid, the standard numerical discretizations of Section 3.1 are used to compute the flow field properties. The following provides a summary of the methods listed above.

3.3.1 Line Methods

This class of interface reconstruction uses line segments to represent various interface configurations in a given cell. It is a true "reconstructive" class in that it uses geometrical and physical arguments (not standard discretization techniques) to solve the volume fraction equation, and explicitly locates and constructs an interface within a cell. Excellent summaries of these methods can be found in Ubbink [49], Kothe and Rider [37], and Rudman [38]. The main reconstruction groups of this method are piecewise constant line segments and piecewise linear segments.

Piecewise Constant Methods: The piecewise constant line segment method was first developed by Noh and Woodward [29] with their SLIC (Simple Line Interface Calculation) method. In this method, the interface in a given cell is reconstructed as a line (in two-dimensions) or a plane (in three-dimensions). Using geometrical considerations based on the volume fraction values of neighboring cells, the line segments are advected in a Lagrangian manner through the flow. The reconstruction method forces the segments to align with one of the coordinate axes of the numerical grid used in the calculation, restricting the SLIC method to structured meshes. Examples of various interface reconstructions for different interface shapes are shown in Figure 3.5.

The SLIC method uses an operator split time integration technique to determine the orientation of an interface, whereby a one-dimensional calculation is repeatedly applied to each coordinate direction in a given time step. Using the reconstruction algorithm briefly described above, the volume fraction information of the (1-D) neighbor cells in a given direction is used to calculate the interface position in a cell. The operator split technique
performs a sweep for each coordinate axis. For example, during an $x$-sweep, cells to the left and right of an interface cell are used to calculate the interface location within that cell, and the interface is aligned normal to the $x$-axis. However, on a subsequent $y$-sweep, cells above and below the interface cell are used to determine the interface location and the line is oriented normal to the $y$-axis. This means that the fluid configuration changes depending on the flow direction and sweep pattern. An example of this directional behavior is shown in Figure 3.6.

Another drawback of the method is that numerical error in advecting the flow causes small pockets of volume fraction (on the order of a typical cell size) to break off from the main flow. This phenomenon is dubbed “Flotsam and Jetsam” [29] and typically occurs when the flow has high vorticity near the interface [37]. This numerical debris makes the accurate determination of interface dynamics with surface tension very difficult, as the calculation must take place on small, highly curved, non-physical regions in addition to the main flow itself. Furthermore, this debris can also lead to spurious velocity fields in the solution.
Overall, piecewise constant line segments are first-order spatially accurate [29, 38]. These reconstruction techniques have fallen out of favour as there are more accurate line methods available with which to reconstruct interfaces. However, due to their relative ease of implementation, they can still be found in limited use today [37].

**Piecewise Linear Methods:** Modern implementations of piecewise linear interface capturing (PLIC) methods are largely based on the work of Youngs [52, 53]. Here a line (2-D) or plane (3-D) is placed within an interfacial cell, and its location is explicitly determined by its slope and intercept with the cell boundaries. The slope of the segment is found through the interface normal vector, which is calculated by examining the volume fraction and fluxes for all cells surrounding (and including) the interfacial cell. The intercept is calculated through volume conservation of the fluid for the given cell. As with the piecewise constant method, geometrical considerations are used to solve the volume fraction equation and locate an interface within a cell. However, the line segment need not be aligned with a coordinate axis, giving more realistic interface approximations. The original method of Youngs uses an operator splitting technique to evaluate the interface configuration in the manner described above. Though Youngs’ original method is only first-order spatially accurate, its results are a drastic improvement over piecewise constant methods due to the capability of reconstructing more realistic interface shapes. This is seen in Figure 3.7, where the fluid configuration of Figure 3.6 is calculated using Youngs’ method. The solution is clearly superior at reproducing the interfacial behavior than the SLIC method. Other interface reconstructions using PLIC techniques are given in Figures 3.8 and 3.9.
Youngs' method is derived solely from a geometrical point of view, and its algorithm cannot be decomposed into a combination of upwind and downwind fluxes (i.e., algebraically) [16, 38, 52, 53]. It is multidimensional in the sense that an interfacial cell needs contributions from all of the nearest neighbor cells in all grid coordinate directions in order to compute the interface. Hence, the original form of Youngs' method is applicable to structured meshes only. Due to its geometrical nature, the method is very complicated to implement in two and three-dimensions. The details of the algorithm and implementation notes are outlined in [16, 37, 38, 52, 53]. Nonetheless, Youngs' method has been modified and extended over the past twenty years, and is a very popular method due to its higher accuracy in reconstructing interface shapes. The original work in [52, 53] has been extended in modern methods to include unsplit time integration techniques, applicability to non-orthogonal and unstructured meshes, and coupling with adaptive mesh refinement procedures. An excellent summary and modification of a modern piecewise linear reconstruction method is given in [37].

Figure 3.8: Interface reconstruction using piecewise linear method (From [10])
Figure 3.9: Comparison of a SLIC reconstruction with a PLIC method for a circle. Local volume fraction values are given and shaded. (From Kothe and Rider [37])
3.3.2 High Resolution Discretization Schemes

An alternative to geometric reconstruction schemes of line segment methods is to reconstruct the interface through a discretization of the scalar transport equation governing the volume fraction. The volume fraction equation can be written in either advective or conservative form

\[
\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla) \alpha = 0 \tag{3.15}
\]

\[
\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \tag{3.16}
\]

Using the conservative form (3.16) (which is more amenable to finite volume discretization), the semi-discrete version of the equation is

\[
\frac{\partial \alpha}{\partial t} + \sum_f \alpha_f (\mathbf{u} \cdot \mathbf{n}_f) A_f = 0 \tag{3.17}
\]

(3.17) is a pure advection equation, and there are several issues to consider when applying standard discretizations to this type of equation. The main difficulty in the numerical solution lies with the discretization of the convective term, \( \nabla \cdot (\alpha \mathbf{u}) \). The volume fraction is an indicator field that represents the ratio of volume of liquid in a given cell to the volume of the cell itself and hence has a physical interpretation. The permitted values of volume fraction for two-phase flow are zero for the gas phase, unity for the liquid phase, and \( 0 < \alpha < 1 \) for cells containing the interface. Values outside of zero and one are non-physical in the sense that they do not represent a realistic case (i.e., there is more volume of a given fluid in a cell than there is volume within the cell itself). Hence, any discretization of (3.17) must satisfy the criteria that solutions be globally bounded by the above condition, and that they be monotonic, i.e., locally bounded (It is reasonable to assume that a solution expected to be locally bounded by two values must then lie within those two values and not obtain any artificial local extrema). Furthermore, the solution must not introduce any numerical diffusion, as we are interested in essentially reproducing a step profile or discontinuity in space, and we want as sharp a profile as possible. Further discussion of issues relating to discretization of convective terms are wonderfully outlined in Leonard [22] and Leveque [25].

The simplest spatial differencing scheme of the convective term is achieved by first-order upwinding. Here, the face values of (3.17) are approximated by the upwind cell value, as in (3.11). It is well known that first-order upwinding provides a solution that is globally bounded and monotonic, but is extremely poor at computing sharp interfaces. The method introduces an excessive amount of numerical diffusion into the solution, as the leading order
error in the discretization is *diffusive* in nature. Higher-order methods such as second-order central and upwind differencing (3.12), (3.13) provide sharper profiles by decreasing the numerical diffusion in the solution, but are especially prone to oscillations near sudden changes (steps) in the solution profile. This is due to the fact that their leading order discretization errors are *dispersive* in nature, and cause spreading of the solution at sudden profile changes. These solutions contain *undershoots* and *overshoots*, which violate the boundedness and monotonicity criteria. An example of this behavior is given in Figure 3.10. Here, various profiles are advected using the above discretization methods and shown to be inadequate in satisfying all the necessary constraints.

![Computation of advected profiles](image)

Figure 3.10: Computation of advected profiles
CHAPTER 3. NUMERICAL METHODS

Clearly, higher order methods increase interface sharpness but add unwanted oscillations. There are many remedies to this problem from the point of view of discretization schemes for convective equations. Certain approaches add numerical diffusion in an attempt to smooth out solutions and preserve monotonicity, then add an appropriate amount of anti-diffusion to cancel out some of the excess and keep a sharp solution while still enjoying the benefits of the added diffusion. These are known as Flux Corrected Transport (FCT) and Total Variation Diminishing (TVD) schemes [31]. These methods add the correct amount of positive or negative diffusion by switching from shape-preserving downwind based discretizations to monotonicity-preserving upwind based discretizations. The approach is to blend different discretizations, using a higher order shape-preserving formulation whenever possible, but then switching to a monotonicity preserving method when certain conditions are violated. The tools used to implement these methods are known as flux-limiters or slope-limiters [24, 25].

TVD limiters are very popular and have been developed for many decades. There are several well known limiters in use today that are applied to convective flows [24, 25]. The idea behind these limiters is to limit the total variation of a numerical solution. The total variation of a solution, $Q_i$, given on a discrete mesh at a time level $n$ is defined as [25]

$$TV(Q) = \sum_{i=-\infty}^{\infty} |Q_i^n - Q_{i-1}^n|$$

(3.18)

A scheme is considered total variation diminishing if

$$TV(Q_{i+1}^n) \leq TV(Q_i^n)$$

(3.19)

This means that solutions can change extremal values, but if the method is TVD, the solution cannot generate new oscillations, as the relative difference between solution extrema cannot increase. A useful conclusion resulting from this property is that any initial monotonic profile advected by a TVD method is free of oscillations and is always monotonic [25].

Many numerical discretization schemes fit into the framework of TVD flux-limiters by defining the following term given by Leveque [25] as

$$\theta_{i+\frac{1}{2}} = \frac{Q_i - Q_{i-1}}{Q_{i+1} - Q_i}$$

(3.20)

This term is a measure of the smoothness in the solution near the grid point $x_{i+\frac{1}{2}}$. For values of $\theta \approx 1$, the solution is considered smooth, while near discontinuities, the solution is not
smooth and \( \theta \) will be far away from unity. We write a generalized flux term at the point \( x_{i+\frac{1}{2}} \) as a 1-D representation of the summation in (3.17) that is a function of the smoothness variable, \( \theta \)

\[
F_{i+\frac{1}{2}} = u \left[ Q_i + \frac{\phi(\theta_{i+\frac{1}{2}})}{2}(Q_{i+1} - Q_i) \right]
\]  

(3.21)

The definition of the function \( \phi(\theta) \) is the key to flux-limited TVD schemes. For a method to be TVD, the function \( \phi(\theta) \) must satisfy the following constraint [25]

\[
\phi(\theta) = \max \left( \min(2, 2\theta), 0 \right)
\]  

(3.22)

We obtain standard discretizations using the above flux formulation by setting \( \phi(\theta) \) to the following values

\[
\phi(\theta) = 0, \text{ 1st-upwinding}
\]  

(3.23)

\[
\phi(\theta) = 1, \text{ 2nd-downwind ("central")}
\]  

(3.24)

\[
\phi(\theta) = \theta, \text{ 2nd-upwinding}
\]  

(3.25)

\[
\phi(\theta) = \frac{1}{2}(1 + \theta), \text{ 2nd-central (Fromm's method)}
\]  

(3.26)

The allowable TVD region and the above discretizations are shown in Figure 3.11. Note that none of the methods with the exception of first-order upwinding satisfy the TVD constraint.

In order to obtain more realistic representations of convected solution profiles, Sweby [21, 25, 44] proposes a more restrictive constraint on the TVD allowable region for second-order methods which prevents the computed profiles from being either too smooth or too compressive. The new allowable region is that of the shaded area in Figures 3.11 b-d. Using this modified TVD diagram, we see that any second-order TVD method must fall within this new shaded region. There are several popular limiters that satisfy these conditions

\[
\phi(\theta) = \max \left( \min(1, \theta), 0 \right), \text{ Minmod}
\]  

(3.27)

\[
\phi(\theta) = \max \left( 0, \min(1, 2\theta), \min(2, \theta) \right), \text{ Superbee}
\]  

(3.28)

\[
\phi(\theta) = \max \left( 1, \min((1 + \theta)/2, 2, 2\theta) \right), \text{ MC}
\]  

(3.29)

Minmod is the most diffusive of the TVD limiters, as it initially follows the slope given by second-order upwinding in the modified TVD region. Use of this limiter will give rounder
CHAPTER 3. NUMERICAL METHODS

Figure 3.11: Allowable TVD regions. a) Shaded TVD area and discretization curves: Beam-Warming (2nd-up), Lax-Wendroff (2nd-down), Fromm. b) Shaded Sweby TVD area. Minmod c) Superbee d) MC (From Leveque [25])

profiles, and will not artificially steepen gradients (turn them into step profiles). In contrast, Superbee is the most compressive of the TVD limiters, as it follows the initial top edge of the allowable TVD region (it has twice the upwind slope here). Superbee has the property that it does cause artificial steepening of any finite gradient into a step. MC (Monotonized Central) is an average of the Minmod and Superbee. It is considered a “safe” limiter for convective flows, as it neither over-compresses nor over-smooths profiles. An example of these limiters is shown in Figure 3.12, in which the same profiles of Figure 3-10 are used for comparison.

As discussed, the use of downwind methods gives sharper profiles, but these are not TVD and hence violate monotonicity constraints. For free surface problems, the method should be as close to downwinding as possible in order to keep tight resolution of interfaces. TVD methods having steeper initial slopes (such as Superbee) which approach the downwind discretization value \( \phi(\theta) = 1 \) as quickly as possible give sharper solutions, while still being TVD and preserving monotonicity. However, according to Leonard, the additional TVD constraints on second-order methods proposed by Sweby are too restrictive [21, 22, 23], and sharper profiles can be obtained if the TVD constraint is relaxed. The limiters can be de-
Figure 3.12: TVD limiters used in advecting various profiles

scribed by a more general class of limiter developed by Leonard, called the Universal Limiter for Tight Resolution and Accuracy (ULTRA) (or the Universal Limiter for short) [21, 23].

Leonard suggests that for a given cell face, the most important information needed in determining a solution face value, \( \phi_f \), due to advection comes from the cells directly downwind, \( \phi_D \), "central", \( \phi_C \) (or just upwind), and "upwind", \( \phi_U \) (two cells upwind from the face) to the given face. A diagram of this naming convention is shown in Figure 3.13. In order to better analyze the concepts developed in [21, 22, 23], we introduce a normalized variable and define the Normalized Variable Diagram (NVD). A normalized variable, \( \tilde{\phi} \), is defined as

\[
\tilde{\phi} = \frac{\phi(x, y, z, t) - \phi_U}{\phi_D - \phi_U}
\]  

(3.30)

As in the case with the smoothness variable, \( \theta \), and generalized flux function, \( \phi(\theta) \), in TVD
methods, we represent several standard discretizations in normalized variable form

\[ \tilde{\phi}_f = \tilde{\phi}_C, \ 1^{\text{st}}\text{-upwind} \]  
(3.31)

\[ \tilde{\phi}_f = \frac{1}{2}(1 + \phi_C), \ 2^{\text{nd}}\text{-downwind} \]  
(3.32)

\[ \tilde{\phi}_f = \frac{3}{2}\tilde{\phi}_C, \ 2^{\text{nd}}\text{-upwind} \]  
(3.33)

\[ \tilde{\phi}_f = \frac{1}{2} + \phi_C, \ \text{Fromm} \]  
(3.34)

These methods are graphically represented on the NVD in Figure 3.14

The TVD methods discussed earlier fall under this method of description, and Minmod
and Superbee are written below. Other limiters are discussed in the framework of the normalized variable in [21]

\[
\tilde{\phi}_f = \begin{cases} 
\frac{3}{2}\tilde{\phi}_C, & \text{if } 0 \leq \tilde{\phi}_C \leq \frac{1}{2} \\
\frac{1}{2}\left(1 + \tilde{\phi}_C\right), & \text{if } \frac{1}{2} \leq \tilde{\phi}_C \leq 1 \\
\tilde{\phi}_C, & \text{if } \tilde{\phi}_C < 0 \text{ or } \tilde{\phi}_C > 1
\end{cases}, \text{ Minmod} 
\] (3.35)

\[
\tilde{\phi}_f = \begin{cases} 
\min\left(2\tilde{\phi}_C, \frac{1}{2}(1 + \tilde{\phi}_C)\right), & \text{if } 0 \leq \tilde{\phi}_C \leq \frac{1}{2} \\
\min\left(1, \frac{3}{2}\tilde{\phi}_C\right), & \text{if } \frac{1}{2} \leq \tilde{\phi}_C \leq 1 \\
\tilde{\phi}_C, & \text{if } \tilde{\phi}_C < 0 \text{ or } \tilde{\phi}_C > 1
\end{cases}, \text{ Superbee} 
\] (3.36)

The graphical representation of these limiters in the NVD is given in Figure 3.15

![Figure 3.15: NVD for Minmod (dashed) and Superbee limiters (solid) (From Leonard [23])](image)

In the context of normalized variables, monotonicity preserving conditions for explicit time advance flows are such that

\[
\tilde{\phi}_C \leq \tilde{\phi}_f \leq 1 
\] (3.37)

\[
0 \leq \tilde{\phi}_U \leq \tilde{\phi}_C 
\] (3.38)

\[
\tilde{\phi}_f \leq \tilde{\phi}_C/c 
\] (3.39)
where $c$ is the Courant number, and the above hold in the monotonic range defined by $0 \leq \tilde{\phi}_C \leq 1$. Outside of this range, the normalized face value can be set to that of first-order upwinding, $\tilde{\phi}_f = \tilde{\phi}_C$, with no loss of accuracy, as the values here are non-monotonic and must be reset to monotonic ones. The method used within the monotonic range determines the overall accuracy of the solution [21, 22, 23]. The conditions, (3.37)-(3.39), constitute the Universal Limiter for a Courant number-dependent flow. The region of validity in the NVD is shown in Figure 3-16.

![Figure 3.16: Universal Limiter for Courant number-dependent flows (From Leonard [21])](image)

The slope in the initial portion of the NVD in Figure 3-16 represents an inverse Courant number dependence. The shaded area is the allowable range for monotonic solutions. In the context of the NVD, the TVD condition proposed by Sweby [44] translates to

$$\tilde{\phi}_f \leq 2\tilde{\phi}_C, \text{ for } 0 \leq \tilde{\phi}_C \leq \frac{1}{2} \tag{3.40}$$

According to the criteria set by the Universal Limiter, this condition is overly diffusive in the first part of the monotonic range, $0 \leq \tilde{\phi}_C \leq \frac{1}{2}$, i.e., the slope is not steep enough precisely where it needs to be a steep as possible in order to produce sharp solutions. Since all TVD limiters obey this constraint, all produce solutions that are more diffusive than needed, according to the Universal Limiter constraints.
CHAPTER 3. NUMERICAL METHODS

The extreme limiting case for monotonic solutions is a method that simply follows the upper boundary of the Universal Limiter. This is known as the Hyper-C method [21] and converts all finite gradients into step profiles. The NVD for Hyper-C is shown in Figure 3-17

Hyper-C is unsuitable for general purpose convection calculations, as it artificially steepens any gradient profile into a step discontinuity. However, this type of behavior is exactly needed for the accurate representation of free surfaces through the discretization of the volume fraction equation (3.16). This method is discussed in further detail in Section 3.4.

Most limiter methods (TVD, FCT, NVD, etc.) are too diffusive to sufficiently capture an interface on a grid, even though there are excellent methods in place for convection dominated flows. Using the Universal Limiter should guarantee monotonic solutions, as it is designed to satisfy exactly this condition. However, most free surface calculations cannot be done simply by using high-order discretizations, as there is a lack of physical information that is needed to accurately advect flow profiles and orient interfaces. Leonard [20, 21, 22, 23], and Leveque [24, 25] offer further details regarding other limiters and higher-order discretizations, and Ubbink [49] gives details concerning higher-order methods used in free surface calculations.

3.3.3 Donor-Acceptor Formulations

Donor-acceptor formulations fall under the general classification of flux-corrected transport (FCT) methods. The algorithms are formulated in an algebraic manner (i.e., using some combination of upwind and downwind fluxes) to prevent overshoots and undershoots in the
interface and provide monotonic surface profiles. Upwind flux values provide artificial di­fusion to the numerical scheme, preventing non-monotonic behavior and giving smoother profiles, and downwind fluxes add negative diffusion in an attempt to keep interface profiles sharp and prevent overshoots and undershoots in the solution near discontinuities (such as interfaces).

The motivation behind donor-acceptor methods is to provide an alternate approach to sophisticated and complicated geometrical line segment interface reconstruction techniques, similar to the philosophy of higher-order discretization methods. They attempt to capture interface orientation through flux correction of the solution to the scalar transport equation governing the volume fraction behavior, (3.16). The most popular version of donor-acceptor formulations is the original VOF method, developed by Hirt and Nichols [15], which was based on the work of Ramshaw and Trapp [36], and summarized by Ubbink [48, 49] and Rudman [38].

The main issue with donor-acceptor formulations is the solution of the volume fraction equation, (3.16), and the resulting problems associated with the discretization of a convective equation, as discussed in Section 3.3.2. Donor-acceptor formulations make use of controlled downwinding [36, 49, 48] to keep interfaces sharp while keeping the volume fraction values within physical bounds. The values in the interfacial “donor” cell and the downwind “acceptor” cell are used to advect the correct amount of fluid through the cell face between them, resulting in physically valid volume fractions when (3.16) is solved. The cell naming convention used here is illustrated in Figure 3.18.

Figure 3.18: Naming convention used for donor-acceptor schemes

If we assume the acceptor cell is completely filled with a uniform fluid, the use of a downwind value to advect the flow and keep interfaces sharp will imply that the acceptor cell will require a volume of fluid equal to the volume of the fluid already contained within it. The donor cell, which contains the interface and hence a combination of both fluids, will
CHAPTER 3. NUMERICAL METHODS
donate the amount of the required fluid that it has available, and then start to donate the remaining volume of the other fluid. In this way, more of the required fluid is not drawn out of the donor than available, preventing the volume fraction values from becoming larger than one or less than zero and representing a non-physical situation. This is the manner in which the VOF method uses downwind values to keep tight interface resolution and keep the volume fraction values bounded. The details are found in [15, 49].

The controlled downwinding of donor-acceptor methods is highly compressive. The use of any amount of downwinding value tends to "wrinkle" interface profiles, turning any finite gradient in the volume fraction into a step profile, even if the gradient is very near to horizontal. As the original VOF method is based on the donor-acceptor formulation, it suffers from this behavior. It is shown by Ubbink [49] that this method is the donor-acceptor equivalent of the Hyper-C high-resolution discretization scheme of Section 3.3.2. In the original VOF method, piecewise constant line segments are oriented parallel to coordinate axes to derive the interface shape, in much the same way as in the SLIC method. However, it allows for "stair-casing" or alignment of the interface with more than one coordinate direction within a given cell, yielding more realistic results than with the SLIC method. An example of this type of reconstruction technique is shown in Figure 3.19.

![Interface reconstruction using donor-acceptor method](image)

Figure 3.19: Interface reconstruction using donor-acceptor method (From Ubbink [49])

Since the interface can be aligned with any coordinate axis, controlled downwinding always orients the line segments vertically due to its compressive nature. In order to remedy this problem, the original VOF method has a built-in control mechanism to determine when the interface should switch from a vertical to a horizontal orientation (in 2-D). The method incorporates all of the cells in the local neighborhood (like Youngs' method) to determine the interface normal vector. The normal is used to change the orientation of the interface
CHAPTER 3. NUMERICAL METHODS

from vertical to horizontal, depending on whether the angle it makes with the direction of motion is greater than 45°. In order to accomplish this, the original VOF method switches from downwinding (acceptor-cell values) to upwinding (donor-cell values), which guarantees bounded results, but avoids the steepening effect that downwind values provide. In this way, the VOF method prevents the interface profile in a cell from becoming a step, and provides a means by which horizontal interface shapes can be attained. The main drawbacks to this remedy are that the switch to upwinding gives a first-order, low accuracy result, that is undesirable (downwinding gives sharper profiles, which are highly desirable if bounded), and the computed profiles exhibit sudden changes from horizontal to vertical alignment, leading to wrinkled, unrealistic interface shapes. This behavior is demonstrated in Figure 3.20.

![Figure 3.20: Interface profile calculation using original VOF method (From Rudman [38])](image)

Newer modified versions of free surface methods based on the original VOF formulation also suffer from this same interface wrinkling issue. Lafaurie et. al. [19] perform studies showing the angle at which the interface orientation changes plays a large role on the accuracy of the results and on the shape of the profile produced. The manner of switching distorts interface shapes, and the distortion varies from method to method. The original VOF and newer methods also suffer from “Flotsam and Jetsam” break-off from the main flow, in the same way the SLIC methods do [38, 49]. Furthermore, as the interfaces are aligned with coordinate axes, the method is limited to structured meshes.

Donor-Acceptor methods are very popular due to their relative ease of implementation, and offer a natural supplement to higher-order methods with respect to free surface simulations. They can keep interface resolutions very tight and provide reasonable results. Some variant of the original VOF method is included in most major commercial CFD codes.
3.3.4 Summary and Conclusions

All of the methods presented above have their own unique advantages and disadvantages with respect to fluid interface reconstruction. Since this project makes use of an existing flow field solver, CMGFD, there exists a certain framework for solution methods that should be taken advantage of. For this and other reasons, the interface reconstruction used in this study takes the approach of propagating the volume fraction field through a discretization of the conservation equation, (3.16), based on the concepts of higher-order discretizations and donor-acceptor formulations. While interface reconstruction by PLIC methods is very sharp and produces good results, their geometric basis yields a very complicated implementation process that does not take advantage of the inherent structure of the CMGFD code. These methods are decided against for interface reconstruction in this study.

Although higher-order methods and donor-acceptor formulations are easier to implement and more compatible within the framework of our existing code, they have several issues concerning the quality of the interface shape produced, as discussed above. A method that takes advantage of the concepts of these two methods and addresses the interfacial shape issues is developed by Ubbink [48, 49] and is known as CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes). This scheme is chosen to perform the free surface reconstruction in this project and the details of this method are discussed below.

3.4 CICSAM method

As mentioned, donor-acceptor schemes make use of controlled downwinding to physically limit the amount of fluid convected in a given time step and keep interface profiles sharp. However, they give rise to unrealistic interface profiles, due to inadequacies in their reconstruction techniques. The CICSAM method is a donor-acceptor formulation that takes advantage of concepts from high-resolution discretization schemes to ensure boundedness and give realistic interface profiles.

The CICSAM method provides sharp interface profiles through a specially designed interpolation of the volume fraction face value, $\alpha_f$, used in the conservative form of the volume fraction equation, (3.17). The calculation procedure used for the volume fraction equation based on the CICSAM method is the same for all faces in the grid, so the details are given only for a single face. The procedure is initially discussed for a 1-D formulation.

As mentioned above, the basis of donor-acceptor formulations is the controlled downwinding procedure, which is the equivalent of the Hyper-C discretization scheme [49]. This
CHAPTER 3. NUMERICAL METHODS

is the most compressive of discretization schemes that satisfies the monotonicity and global boundedness criteria outlined in Section 3.3.2. In terms of normalized variables, the normalized face value for this scheme (using the naming convention in Section 3.3.2) is given by

\[ \tilde{\alpha}_{HC} = \begin{cases} \min \left( 1, \frac{\alpha C}{\alpha I} \right), & \text{if } 0 \leq \tilde{\alpha}_C \leq 1 \\ \alpha C, & \text{if } \alpha C < 0 \text{ or } \alpha C > 1 \end{cases} \] (3.41)

Here, \( c_t \) represents the total outflow Courant number for a given cell. This is calculated by summing all of the outflow Courant face values for the cell

\[ c_t = \sum_f \max \left( \frac{(u_f \cdot n_f)\Delta t}{\Delta V}, 0 \right) \] (3.42)

where \( n_f \) represents the outward normal to a cell face \( f \), \( u_f \) is the velocity at the given face, \( \Delta t \) is the time step used in the calculation, and \( \Delta V \) is the volume of the given cell.

The main drawback of the Hyper-C method, (3.41), is that the interface profiles produced are wrinkled, i.e., they will always be turned into vertical step profiles. In order to generate more realistic surfaces, CICSAM blends different discretizations to give different interface orientations. The main measure of surface orientation is the interface normal vector, \( \hat{n} \)

\[ \hat{n} = \frac{\nabla \alpha}{|\nabla \alpha|} \] (3.43)

The gradient terms, \( \nabla \alpha \), are calculated using a generalized form of Gauss' Theorem [42, 49]. The gradient is treated as a special form of a divergence operator, and when taking the volume integral of a given cell, the gradient is represented as

\[ \int \int \int_{\Delta V} \nabla \alpha dV = \oint_{\Delta A} \alpha n dA \] (3.44)

In turn, the surface integral is simply the summation of the face values of the given quantities in (3.44) over the total number of faces in the given cell

\[ \oint_{\Delta A} \alpha n dA = \sum_f \alpha f n_f A_f \] (3.45)

The original VOF method avoids the constant step profiles produced with the controlled downwinding method by switching from a downwind face approximation to an upwind value...
for \( \alpha_f \) when the interface normal is more than 45° above horizontal. This produces solutions that have sudden switches in topography from horizontal to vertical. Other updated donor-acceptor variants [19] perform studies showing the choice of the angle in the switching criteria greatly affects the final shape of the interface. In order to remedy this problem, the CICSAM method gradually blends the Hyper-C formulation (3.41) with an appropriate upwinding scheme. The Hyper-C face term, \( \tilde{\alpha}_{fHC} \), is used to preserve shape when the interface is very close to a step profile normal to the flow direction. The upwind face value, \( \tilde{\alpha}_U \), is switched to if monotonicity must be preserved and is dominant when the interface is nearly horizontal or essentially tangential to the flow. The blended, normalized face value, \( \tilde{\alpha}_f \), is given as

\[
\tilde{\alpha}_f = \gamma_f \tilde{\alpha}_{fHC} + (1 - \gamma_f) \tilde{\alpha}_U
\]

The face value is a linear blend of the two terms that depends on the blending constant, \( \gamma_f \). This term is given as

\[
\gamma_f = \min \left( k, \frac{\cos(2\theta_f) + 1}{2}, 1 \right)
\]

\( k \) is a constant used to control the dominance of the different schemes. For a value of \( k = 0 \), the upwind term remains, while for very large values, the Hyper-C term dominates. The recommended value is \( k = 1 \). \( \theta_f \) describes the angle between the cell interface normal vector, \( \nabla \alpha_c / ||\nabla \alpha_c|| \), and the vector connecting the centers of the donor and acceptor cells. The CMGFD code uses structured meshes, so this is simply the unit normal vector to the cell face, \( n_f / ||n_f|| \)

\[
\theta_f = \cos^{-1} \left( \frac{\nabla \alpha_c \cdot n_f}{||\nabla \alpha_c|| n_f||} \right)
\]

In this way, CICSAM blends the Hyper-C scheme with an upwind scheme to generate realistic interface profiles. As noted by Ubbink [48, 49], the original VOF method switches to a first-order upwind scheme. While first-order upwinding is bounded and monotonic, it does a horrible job at keeping interface resolution. Hence, a more appropriate upwind scheme should be switched to, i.e., one that does a better job of maintaining the interface sharpness. Based upon studies performed by Leonard [21], CICSAM switches from Hyper-C to the ULTIMATE-QUICKEST scheme [20, 21]. This is a third-order upwind scheme that is a transient version of the original QUICK third-order method [20], bounded by the Universal Limiter. The upwind face value for ULTIMATE-QUICKEST is given in terms of normalized
CHAPTER 3. NUMERICAL METHODS

variables

$$\tilde{\alpha}_U = \left\{ \begin{array}{ll}
\min \left( \frac{\eta \tilde{\alpha}_C + (1-\eta) (6 \tilde{\alpha}_C + 3)}{8}, \tilde{\alpha}_{fHC} \right), & \text{if } 0 \leq \tilde{\alpha}_C \leq 1 \\
\tilde{\alpha}_C, & \text{if } \tilde{\alpha}_C < 0 \text{ or } \tilde{\alpha}_C > 1
\end{array} \right. \quad (3.49)$$

Using the blended formulation of the normalized face value, (3.46), with the appropriate upwind and downwind terms, (3.41), (3.49), we determine the actual face value, $\alpha_f$, for use in the semi-discrete volume fraction equation, (3.17). Using the definition of the normalized variable, (3.30), we re-arrange the normalized face value for the actual face value, $\alpha_f$

$$\alpha_f = (1 - \beta_f) \alpha_C + \beta_f \alpha_D \quad (3.50)$$

The weighting factors, $\beta_f$, are given by

$$\beta_f = \frac{\tilde{\alpha}_f - \tilde{\alpha}_C}{1 - \tilde{\alpha}_C} \quad (3.51)$$

It is clear from (3.50) that the face values, which contain all of the information needed to keep sharp interface profiles through the weighting factor, $\beta_f$, are written as a linear combination of the cells straddling a given face. Also, the weighting factors implicitly contain information concerning the upwind cell, $\alpha_U$, through the normalized value of the donor cell, $\alpha_C$. This upwind information is helpful in preventing artificial steepening of volume fraction gradients, and helps prevent unrealistic profiles from occurring when using the Hyper-C method [48, 49].

The derivation of the CICSAM method has so far taken place within a one-dimensional context. However, the CMGFD code is fully three-dimensional, so the question of how to extend this method to several dimensions is posed. As the original CICSAM is intended for arbitrary meshes, great care is taken to avoid a procedure that relies on an operator splitting technique, i.e., one which repeated a given 1-D calculation over each coordinate direction in the grid. Ubbink devises a flux-averaging scheme based on the Crank-Nicolson time integration method, using the face values of the past, $\alpha_f^n$, and current, $\alpha_f^{n+1}$, time levels to approximate the current face value, $\alpha_f^{n+1}$ [48, 49]. Since our implementation uses structured meshes, the concept of operator splitting is inherent to the CMGFD code itself, so the averaging procedure is included in order to flux the proper volume of fluid across a face in several dimensions. The 1-D calculation used to find $\alpha_f$ in (3.46) is repeated over each coordinate direction, and the flux-averaging procedure, which is embedded within the Crank-Nicolson method, is automatically used during the solution of the volume fraction equation. The details of this procedure are straightforward and discussed in [48, 49].

A multi-dimensional extension involves the Courant number dependence of the CICSAM
method. In (3.42), the total outflow Courant number for a given cell is calculated, implying that all cell faces are considered within a given step and used by the CICSAM method. This inclusion of all cell faces incorporates the influence of all spatial dimensions in the evaluation of the face values, giving it further and more accurate multi-dimensional behavior.

The discussion of the CICSAM scheme has hitherto been concerned with the spatial discretization of the volume fraction equation, with the resulting semi-discrete form given in (3.17). The details of temporal discretization are now addressed. The semi-discrete form is integrated in time to generate several different time discretization methods. As the CICSAM method is time sensitive through its Courant number dependence, we employ a higher-order temporal discretization to retain the accuracy of the volume fraction solutions during the time stepping procedure. The CMGFD code is implicitly discretized in time, so it is only logical that the volume fraction equation should also be discretized implicitly in time as well. Since the operator splitting technique is to be adopted, the flux-averaging procedure and the Crank-Nicolson time integration scheme are necessary components of the CICSAM method [48, 49]. The Crank-Nicolson scheme is a second-order accurate implicit method. The semi-discrete equation is integrated over a time interval, Δt, and the trapezoidal method is used to evaluate the resulting integral. The method blends an equal amount of information from the past and current time steps, resulting in a fully discrete equation of the form

\[ ΔV \frac{α_C^{n+1} - α_C^n}{Δt} + \frac{1}{2} \sum_f (α_f^{n+1} u_f^{n+1} + α_f^n u_f^n) \cdot n_f A_f = 0 \]  

(3.52)

As an alternative to the Crank-Nicolson method, another second-order accurate scheme is the three time-level method [9, 28]. Here, the time derivative term is integrated over a time interval, Δt, centered about the current level, n. The midpoint rule is used to approximate the integral, and a parabola is used to interpolate the solution. The method is more straightforward to implement than the Crank-Nicolson scheme. The fully discrete version of the volume fraction equation is given as

\[ \frac{3α_C^{n+1} - 4α_C^n + α_C^{n-1}}{2Δt} + \sum_f α_f^{n+1}(u_f^{n+1} \cdot n_f)A_f = 0 \]  

(3.53)

While this method is second-order accurate, it has a greater error than Crank-Nicolson [9]. The three-time level method does not incorporate the flux-averaging procedure, hence one can test the effect of omitting this feature of CICSAM. The fully discrete equations, (3.52), (3.53), are solved for the current volume fraction field, α(x, t)^{n+1}. The summation in each
equation occurs over all 6 faces of a three-dimensional cell and the equation is solved using an iterative line Gauss-Seidel method as discussed in [13, 14, 30].

The CICSAM interpolation depends on the Courant number in a given cell to maintain sharp profiles through the use of Universal Limiter constraints [21, 49]. In order to keep the method stable, the cell Courant number must be kept below \( c_t = 1 \), (i.e., the CFL condition). According to Ubbink [49], the CICSAM method gives diffusive results for Courant numbers larger than approximately \( c_t = 0.5 \). To achieve optimal resolution in the solution profiles, the Courant number must be kept below this value. During the early stages of iterative solution in the CMGFD code, cell Courant numbers much greater than \( c_t = 0.5 \) occur during the “inner” iterations. These large values can corrupt the CICSAM method and render it useless for computational purposes, as the stability limit of \( c_t = 1 \) for explicit flow discretizations is violated. A natural remedy to this problem, by definition, is to reduce the time step in an attempt to reduce the Courant number. However, this approach leads to very small time steps and gives prohibitively long run times when computing in an implicit manner. An effective solution to this problem is through the use of an implicit-explicit (IMEX) flux formulation. During the inner iterations, an implicit, non-Courant number dependent method is used for the interpolation of face values until later in the iterative process when the Courant numbers naturally reduce through iteration and the CICSAM method is used. There are several effective IMEX formulations in the literature [1], but the one chosen for our purposes is based on a finite-volume formulation for general purpose advection flows, developed by Dendy et. al. [6]. The flux value at a given face is the combination of an implicit and explicit face value

\[
\alpha_f = \alpha_{f,IM} + \alpha_{f,EX}
\]  

(3.54)

The amount of each value used is dependent upon a maximum Courant number threshold, \( c_{\text{max}} \), set by the user. The implicit face value is given by

\[
\alpha_{f,IM} = \max\left(0, 1 - \frac{c_{\text{max}}}{c_t}\right) \alpha_{f,SS}
\]  

(3.55)

The term \( \alpha_{f,SS} \) refers to an implicit formulation used for interpolating the face value. The authors in [6] originally use simple first-order upwinding, but in the interest of preserving interface profiles, we employ a sharper method. The present work uses the Superbee face values for the implicit face term in (3.55).
The explicit contribution to the face value is represented as

\[
\phi_{f,EX} = \min\left(1, \frac{c_{\text{max}}}{c_t}\right) \left[ \min\left(1, \left(\frac{c_t}{c_{\text{max}}}\right)^\beta\right) \alpha_C \right] + \left[1 - \min\left(1, \left(\frac{c_t}{c_{\text{max}}}\right)^\beta\right) \alpha_{f,\text{CICSAM}} \right]
\]

In this term, \(\alpha_C\) refers to the donor cell volume fraction value. For Courant numbers near the prescribed threshold \(c_{\text{max}}\), (3.56) is weighted so that the explicit term will use the value of the first-order upwind cell within a given range from this threshold. The parameter \(\beta\) controls the smoothness of the change from the donor cell value to the CICSAM face value. According to the suggestion in [6], the value of \(\beta\) is set to 50, so that transition occurs smoothly between the two values from \(c_{\text{max}} - 0.05\) to \(c_{\text{max}}\). A value of \(\beta = \infty\) gives a step profile change at \(c_{\text{max}}\) from CICSAM to donor cell.

The value \(c_{\text{max}}\) is arbitrarily set and determines the blending between the implicit and explicit contributions in (3.54). Numerical experimentation shows the Superbee method gives sharper profiles than CICSAM for Courant numbers greater than 0.5. Hence, \(c_{\text{max}}\) is set to 0.5, and an increasing Courant number beyond this level will use a blend of Superbee and first-order upwind face values, with diminishing weight being given to the upwind term during the increase. In this manner, the inner iterations of the solution profile will not give rise to an instability in the face values, and once the Courant number has naturally subsided during the process of iteration, the IMEX method will automatically use the CICSAM method to compute the face values needed for solution of the discrete equation, (3.53).

With all of the above techniques in place, we have the means to solve the volume fraction equation through a discretive technique. It is interesting to note that an operator splitting method cannot truly limit convective processes in a multi-dimensional sense. In an excellent paper, Thuburn [45] discusses and demonstrates that an operator split application of the Universal Limiter (i.e., applying the 1-D version of the limiter to each coordinate direction separately) is not successful at limiting multi-dimensional flows. The procedure for generating face values with the CICSAM method provides solutions that are very sharp, however, they cannot truly guarantee bounded volume fraction values. Ubbink offers a correction procedure for non-physical values by deriving a “predictor-corrector” scheme [48, 49]. The predictor stage is the computation of the volume fraction face values as described above, while the corrector scheme applies a correction to those cells with non-physical volume fraction values. During this correction scheme, the entire system of equations must be resolved until the volume fraction value is properly reduced. This can give rise to the solution of the equations many times within a single time step, drastically increasing the run time of the program. However, Ubbink reports that the non-physical values obtained through the use of
the CICSAM with the flux-averaging are minor and tolerable for a regular mesh \cite{49}. The second-order time integration method with the IMEX face formulation should allow accurate computation of free surface profiles without large oscillations in the computed values. However, numerical experiments show that non-physical values do arise in the absence of the corrector stage.

Rather than implementing the iterative correction procedure, motivation is taken from Dendy et al., whereby they use an \textit{a posteriori} application of a multi-dimensional limiter proposed by Thuburn \cite{45} on the volume fraction field after the solution process. This application guarantees physical solution values for the volume fraction without the use of the repeated correction procedure. However, Thuburn’s limiting procedure is quite involved, and is not easily compatible with the \textit{CMGFD} code. Instead, Leonard \cite{22} notes that Thuburn’s limiter is an extension of the Universal Limiter, and proposes a simple multi-dimensional modification. The modified Universal Limiter follows the same monotonicity bounds as in (3.37)-(3.39), but uses the \textit{total} outflow Courant number for a given cell, (3.42), as opposed to the Courant cell face values, as in the one-dimensional case \cite{21, 22}. The resulting NVD diagram is slightly modified from that of Figure 3.16, in that the Courant number dependent slope is less steep than for the 1-D version. The modified NVD is given in Figure 3.21. This modification allows for the inclusion of multi-dimensional information into the limiting process, and its application to the volume fraction field provides a simple alternative to the methods proposed by Thuburn and Ubbink for the prevention of non-physical values. Therefore, the current work makes use of the multi-dimensional version of the Universal Limiter applied to the volume fraction face values in order to preserve monotonicity in the solution. Numerical tests show that the use of this limiter with the CICSAM method gives volume fraction values that are bounded to within small fractions of a percent in most cases. Hence, we have an effective method by which to calculate and limit volume fraction values for the computation of free surface profiles.

As a final note, Ubbink performs several comparison tests of the CICSAM method with other popular, well established methods \cite{48}. The CICSAM method is shown to greatly outperform the original VOF formulation, the SLIC method, and a FCT method. Only the method of Youngs outperforms CICSAM (and then only slightly) in the test cases. However, as previously mentioned, Youngs’ method is based on geometrical considerations and is very difficult to implement, while the CICSAM method is simply based on the discretization of the volume fraction equation, and is substantially simpler to implement. Therefore, CICSAM offers a more straightforward approach to free surface tracking, while using a much less complex numerical implementation. Results obtained with this method are very good,
and only slightly less accurate than those of PLIC methods, which are considered highly accurate in reconstructing free surface interface profiles.

3.5 Surface Tension Implementation

As described in Section 2.2.3, surface tension effects are implemented into an interface capturing method by means of the Continuum Surface Force (CSF) method of Brackbill et. al. [3]. In this method, surface tension contributions to the interface boundary conditions are replaced by a volumetric force acting within a thin transition region around the interface location. The force is non-zero within this region, mimicking the surface tension effects, and zero elsewhere throughout the numerical grid. In the limit of a vanishing transition thickness, the volumetric force returns the correct value of the surface tension boundary condition. This method of dealing with interfacial forces falls under the larger class of problems known as Immersed Boundary Methods, developed by Peskin [35]. The details of this concept in relation to surface tension flows are given in Section 2.2.3, and the issues regarding its numerical implementation are discussed below.

For a constant surface tension coefficient, \( \sigma \), the interfacial stress due to surface tension is given by

\[
F_{\sigma_A} = \sigma \kappa \mathbf{n}
\]  

(3.57)
CHAPTER 3. NUMERICAL METHODS

The curvature of the interface is \( \kappa \), and the unit normal vector to the interface is \( \hat{n} \). The volumetric force of surface tension acting within the transition region, \( \Gamma \), is represented as

\[
F_{\sigma_v} = \iiint_{\Gamma} F_{\sigma_A} \delta(x - x')dV
\]  

(3.58)

The delta function, \( \delta(x - x') \) restricts the volumetric force to points on the surface, \( x' \), which are located within \( \Gamma \).

There are several issues to consider when calculating surface tension effects. The foremost concern is with the transition region and calculation of the curvature. The curvature of a surface is found by taking the divergence of the unit normal vector to the surface

\[
\kappa = -\nabla \cdot \hat{n}
\]  

(3.59)

The unit normal to the surface is represented as:

\[
\hat{n} = \frac{\nabla \alpha}{|\nabla \alpha|}
\]  

(3.60)

Therefore, the curvature is essentially written as

\[
\kappa = -\nabla \cdot \left( \frac{\nabla \alpha}{|\nabla \alpha|} \right)
\]  

(3.61)

(3.61) states that the curvature is a second spatial derivative of the volume fraction, which must be twice differentiable for accurate curvature calculation. Ideally, the volume fraction is a step profile near the interface, and therefore is not twice differentiable and not appropriate for use in the calculations. The remedy is to define a smoothed volume fraction field, \( \bar{\alpha}(x, t) \), that is twice differentiable and makes a smooth transition from different volume fraction values within \( \Gamma \). The smoothed field is used in place of the original volume fraction field for curvature calculations. An example of the smoothing is seen in Figure 3.22, where an initial circular step profile is smoothed using an appropriate technique.

There are several methods available by which to carry out the smoothing procedure. The original work of Brackbill et. al. [3] uses a B-spline, \( B(x - x') \), to define their smoothed volume fraction field

\[
\bar{\alpha}(x) = \iint_{\Delta V} \alpha(x')B(x - x')dV'
\]  

(3.62)
Further work by Williams et al. [51] extends this idea to general smoothing kernels, \( K(x - x') \), that are convoluted with the volume fraction field to produce the smoothed volume fraction field.

\[
K * \alpha(x) = \bar{\alpha}(x) = \int \int \int \alpha(x') K(x - x') dV'
\]  

(3.63)

The idea of using smoothing kernels is very popular and is used in many recent research projects involving interfacial dynamics [4, 39]. Motivated by the original work of Peskin [35] and a more recent implementation by Bussmann [4], a version of the convoluted field based on Peskin's filter is implemented into CMGFD. Since our code uses an operator splitting technique, Peskin's original method is followed and the kernel used is a product of three one-dimensional delta functions

\[
K(x - x') = \delta(x - x') \delta(y - y') \delta(z - z')
\]  

(3.64)

Each one dimensional delta function is constructed based upon a specially defined cosine function [35] (graphically represented in Figure 3.23)

\[
\delta(x - x') = \begin{cases} 
\frac{1}{4 \Delta x} \left( 1 + cos\left( \frac{\pi(x-x')}{2 \Delta x} \right) \right), & \text{if } |x - x'| \leq 2\Delta x \\
0, & \text{if } |x - x'| > 2\Delta x 
\end{cases}
\]  

(3.65)

The method is implemented for non-uniform structured grids, and the details of this
process are briefly outlined in [4]. This specific convolution filter is nicely incorporated into the framework of the CMGFD program due to its separation into coordinate axes, and is widely used by many researchers to simulate very complex flow processes. In specific, the works of Bussmann, Rudman and Pasandideh-Fard et al. [5, 32, 33, 39] give an example of the quality of result that is obtained with this type of method. A variety of other filters are proposed and summaries of different methods are given in [18, 51].

An alternative to the use of convoluted filters is that of Laplacian filters [17, 19, 49]. Here, a simple averaging process is used to sufficiently smooth out the volume fraction field. The formulation given by Ubbink [49] is also implemented into the CMGFD code for greater robustness in calculating smoothed volume fraction profiles. The smoothed volume fraction field, \( \hat{\alpha} \), is obtained by repeatedly applying (if desired) the Laplacian filter, \( \mathcal{F}(\alpha) \), to the original volume fraction field, \( \alpha \)

\[
\hat{\alpha} = \mathcal{F}^{(m)}(\alpha)
\]

where \( m \) represents the number of applications of the filter to the volume fraction field (the suggested number is \( m=2 \) [49]). The actual filter, \( \mathcal{F}(\alpha) \), is defined as

\[
\mathcal{F}(\alpha) = \frac{\sum_f \alpha_f |n_f A_f|}{\sum_f |n_f A_f|}
\]
CHAPTER 3. NUMERICAL METHODS

This filter redefines the volume fraction value in each individual cell of the computational grid. The face values, \( \alpha_f \), obtained by the CICSAM or Superbee methods are too sharp for use in this procedure. Therefore, \( \alpha_f \) used in (3.67) is obtained through "central" or downwind based differencing, (3.12). This provides a sufficiently diffusive result, which is necessary for proper smoothing of the field. In order to use the filter (3.67) in repeated applications, the original volume fraction field is first smoothed, and then the filter is applied to the smoothed volume fraction in each subsequent pass.

With two different methods in place for smoothing the volume fraction field, namely (3.63) and (3.67), accurate calculation of the curvature takes place using the smoothed field, \( \bar{\alpha} \). There are different approaches by which to calculate the curvature according to (3.61). One method is to evaluate the equation using standard discretization techniques. This method is used by Brackbill et al. in the original CSF method [3]. Here, the interface normal, \( \hat{n} \), smoothed with the Laplacian filter is obtained in the same manner as in (3.45) by using a generalized form of Gauss' theorem. When evaluating the integration in (3.58) numerically, the midpoint rule is used for quadrature, and the value of \( \mathbf{F}_{\sigma_A} = \sigma \kappa \hat{n} \) is taken to lie at cell centers. This is a second order approximate result, and is used for the terms in the surface tension calculations. This means the volume integral of the term \( \mathbf{F}_{\sigma_A} \) is represented by the values of each of the constituent terms at the cell centers

\[
\mathbf{F}_{\sigma_V} = \int \int \int_{\Delta V} \mathbf{F}_{\sigma_A} \delta(x - x') dV = \sigma \kappa P \hat{n}_p V_p \sum_{x,y,z} \delta(x - x')
\]

The methods presented here can be used with non-uniform grids, as the summation in (3.68) simply extends over all cells within the given range of influence of the kernel used. This means that different numbers of cells can be included in the calculation, depending on the mesh density. The term \( \kappa_P \) is found by taking the volume integral of the expression in (3.61)

\[
\kappa_P = - \int \int \int_{\Delta V} \nabla \cdot \left( \frac{\nabla \bar{\alpha}}{\| \nabla \bar{\alpha} \|} \right) dV = - \oint_{\Delta A} \left( \frac{\nabla \bar{\alpha}}{\| \nabla \bar{\alpha} \|} \right) \cdot \mathbf{n} dA = - \sum_f \left( \frac{(\nabla \bar{\alpha})_f}{\| \nabla \bar{\alpha} \|_f} \right) \cdot \mathbf{n}_f A_f
\]

In order to determine the value of the volume fraction gradient at the face, central differencing is used for the interpolation.

An alternative approach to the evaluation of the curvature is to use the convolution method to determine the interface normal, \( \hat{n} \). The spatial gradient of the volume fraction
CHAPTER 3. NUMERICAL METHODS

Field is represented as the convolution of \( \alpha \) with the derivative of the kernel used [3, 51]

\[
\nabla \tilde{\alpha}(x) = \iiint_{\Delta V} \alpha(x') \nabla K(x' - x) dV'
\]

(3.70)

This method offers an alternative approach to calculation of interface normals using Gauss' theorem (3.45). While the convolution method is attractive, one must note that in order to resolve the derivative of the kernel accurately, there must be a relatively large number of grid cells in the region of the kernels' (compact) support to obtain a correct result [51]. Any higher derivatives beyond the first derivative will become singular at a very high rate, and accurate evaluation of these terms will require a prohibitively excessive number of grid cells. Hence, the computation of first derivatives (interface normals) may take place using (3.70), but computation of higher derivatives using this method should be avoided [51]. Therefore, convolution based methods for direct evaluation of the second spatial derivatives in (3.61) should not be used. After calculating the interface normals by means of (3.70), the calculation procedure of (3.69) is performed to evaluate the curvature of the interface.

The final term of importance in the volumetric representation of the surface tension force, (3.68), is the delta function term, \( \delta(x - x') \). This function is used to restrict the volumetric force to the transition region about the interface. The original CSF method uses the magnitude of the volume fraction as the restricting function

\[
\delta(x - x') = |\nabla \tilde{\alpha}|
\]

(3.71)

Inclusion of this term restricts surface tension effects to an area given by the domain of \(|\nabla \tilde{\alpha}|\). It is worth noting that the extent of this magnitude term depends on the amount of smoothing used for the volume fraction term. An unsmoothed term has a profile that is very near to a step profile, hence the gradient term should only be non-zero and a very thin region near this step. Smoothing increases the extent of this domain by widening the step, hence increasing the range of influence of the force.

Once the volume force term, \( F_{\sigma V} \), is calculated from (3.68), it is multiplied by a density factor in order to convert it into a proper body force (this will cause denser fluid elements to experience the same acceleration as lighter ones [3, 17, 18]). The final form of the surface tension term for inclusion into the momentum equations is in general given by

\[
F_{\sigma} = \frac{\rho F_{\sigma V}}{\frac{1}{2}(\rho_l + \rho_g)}
\]

(3.72)
where $\rho$ is the composite density (2.12), and $\rho_l, \rho_g$ represent the density values of the liquid and gas phases respectively. While there is still debate as to whether or not the surface tension contribution should be a true body force, the results given by Kothe et al. [17] show that this density factor is essential to obtaining accurate surface tension results for high density ratios, and therefore the term is adopted in this study.

The evaluation of the surface force by (3.68) is accomplished by the methods described above. Some authors also apply their smoothing filters to the final source term, $F_{\sigma v}$ [4, 39]. While this could be done in principle, it is best to avoid using excessive smoothing, as it should only be minimally incorporated to achieve sufficiently smooth data for curvature calculation. Unfortunately there is no “rule of thumb”, except for perhaps the less smoothing used, the better. The aforementioned techniques provide a unique method of applying surface tension forces to an interface tracking method.

### 3.6 Chapter Summary

This chapter discusses a number of numerical methods used in the calculation of free surface profiles. In the interest of time affordability and ease of implementation into an existing flow field solver, the interface capturing class is used, and the fluids are combined into a single discontinuous composite fluid. A summary of different interface reconstruction techniques is presented, and the CICSAM high-resolution discretization method is chosen for implementation into the existing code. Surface tension effects are incorporated by means of the Continuum Surface Force method. The current implementation has full three-dimensional free surface capabilities within the existing computer code.
Chapter 4

Validation and Test Results

Upon implementing the proposed methods of computing free surface profiles and surface tension effects into the CMGFD code, the results are checked for correctness and validity. This chapter presents the results of a series of test cases designed to verify the proper functioning of the modified computer program.

4.1 Advection Test Cases

These test cases are used by Rudman [38] and Ubbink [48] to compare the performance of several interface capturing methods, and provide a quantitative basis for the verification of the results obtained using our code. They are designed to check the validity of a given method through advection of shapes across a computational mesh over several time steps. Analytical solutions for these simple cases of advection are known, meaning direct error estimates are possible. All of the cases are two-dimensional and use uniform structured meshes. The tests use constant velocity fields to advect shapes in different manners, so the momentum equations are not solved, and there is no coupling of the volume fraction to the velocity and pressure. The tests only provide an estimate of the degree of accuracy of the solution method used for the volume fraction equation.

4.1.1 Constant, Oblique Velocity Field

A unidirectional velocity field, $v = < 2, 1 >$, is used to advect different volume fraction fields across a uniform mesh of size $x, y \in [-2,2]$, consisting of 200 x 200 cells, with a Courant number of 0.25. The shapes used are a hollow square aligned with the coordinate axes, and a hollow circle. The outer and inner lengths of the square are 40 cells and 20 cells.
respectively, while the inner and outer radii of the circle are 10 cells and 20 cells respectively. The centers of each shape are located at (-1.2, -1.2), and the simulation is run for 500 time steps. The tests are performed separately for each of the fields described above.

According to the volume fraction conservation law, advecting any profile in a constant velocity leaves it unaltered. We simply translate the shape to the expected location and use it as the exact solution for purposes of error estimation. The error in the solution is given by Rudman [38] as

$$E = \frac{\sum_{i,j} |\alpha_{i,j}^n - \alpha_{i,j}^e|}{\sum_{i,j} \alpha_{i,j}^0}$$  \hspace{1cm} (4.1)$$

where $\alpha_{i,j}^n$ is the computed solution, $\alpha_{i,j}^e$ is the exact solution, and $\alpha_{i,j}^0$ is the initial solution profile. The error values obtained are used to directly compare the performance of the current CICSAM implementation in the CMGFD code to that of the original CICSAM method and other reconstruction techniques as given in [48]. Figure 4.1 shows the initial volume fraction fields and the results of the advection tests.

Qualitatively, the current CICSAM implementation does a good job in preserving the square profile through many advection steps. This is expected as the square has its sides aligned with coordinate axes, and operator split methods can more easily deal with profiles of this type. Figure 4.1b shows that the main differences in the computed profile versus the exact solution are a slight spreading of the interface, slight diffusion in the direction of advection, and a rounding of the corners. The spreading and diffusion are expected after advecting the profile over many steps, and the rounding of the corners is due to the fact that a corner represents a singularity on an Eulerian mesh, and all interface tracking methods have difficulties in resolving this feature [38].

The circular shape is less faithfully computed, as seen from Figure 4.1d. The differences in the computed and initial results are a smoothing of the jagged initial profile, a slight spreading of the interface, and a distortion in the direction of advection. Since the circle is not aligned with any particular coordinate axis, it is less amenable to operator splitting, so one might expect the current method (even with the Crank-Nicholson averaging) to experience more difficulty in advecting such a shape. The results in [38, 48] show that other interface capturing methods also have a higher degree of difficulty in advecting circular profiles in comparison to square profiles. This lack of alignment with a coordinate axis causes a bias towards the advection direction to appear as a ripple in the circular profile.

Furthermore, the lack of Ubbink’s corrector step may also have an influence on our results. Numerical experiments show that without the use of the Universal Limiter or any correction...
procedure, advection of the circular profile produced unacceptable overshoots in the solution profile, of approximately 3-4% in size at a Courant number of 0.25. The solutions presented above make use of the limiter and not the correction procedure, and this most likely gives rise to a difference in the results presented here when compared with those of Ubbink [48]. With the exception of the ripple in the circular profile, the computed results do qualititatively agree with those presented in [38, 48].

<table>
<thead>
<tr>
<th></th>
<th>SLIC</th>
<th>Hirt-Nichols</th>
<th>FCT-VOF</th>
<th>Youngs</th>
<th>Ubbink</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square</td>
<td>1.32E-01</td>
<td>6.86E-03</td>
<td>1.63E-08</td>
<td>2.58E-02</td>
<td>2.50E-02</td>
<td>3.45E-02</td>
</tr>
<tr>
<td>Circle</td>
<td>9.18E-02</td>
<td>1.90E-01</td>
<td>3.99E-02</td>
<td>2.98E-02</td>
<td>4.33E-02</td>
<td>8.33E-02</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of error values for translation tests
Quantitatively, as indicated in Table 4.1, the current method does fall within the range of errors of CICSAM as presented by [48] but underperforms slightly. However, our results are better, as expected, than those of the other entries in Table 4.1, within the limitations discussed in [38]. It is important to note that the solution method and code structure used by Ubbink [48, 49] is different than the type in our code, so we expect differences in error values for solutions to the same problem. The difference in error in the square profile is relatively small and seems reasonable if we consider to arise from this difference in solution methods and implementation procedures.

The larger error value for the circle is foremost due to the presence of the ripple in the final profile, but also to the fact that the jagged profile is used as the representation of a circle for an initial shape. Within the CMGFD code, there is no control over the initial interface orientation within a given cell. This jagged profile is the discrete analogue to a circle for the purposes of initial conditions and is not very smooth or continuously varying. The CICSAM method however is very adept at producing smoothly varying profiles, and tries to interpret the changes in the interface normal present in the jagged profile as corresponding to those of a circle, as intended. The diffusion due to a large number of advection steps coupled with this increased reconstruction capability gives rise to an inherent difference in solution shape whenever this initial circular profile is used. Regardless of the presence of the ripple, this automatically gives a greater difference in profiles, leading to an increased error. It is safe to say that these two issues account for the difference in values presented for the circle in Table 4.1.

The solutions obtained by the current method are quite well resolved and are in good qualitative and quantitative agreement with available results. These results show that our implementation of CICSAM handles translation tests well.

4.1.2 Solid Body Rotation

This test exposes a volume fraction field in the shape of a slotted circle to a velocity field corresponding to solid body rotation \( \mathbf{u} = < \omega r \sin \theta, -\omega r \cos \theta > \), as shown in Figure 4.2. Here \( \omega \) represents the angular velocity of the field, and is chosen such that the velocity has a magnitude of 1.0 at the center of each edge of the domain. The same mesh resolution is used as in the oblique velocity advection test. The circle is centered at (0.0, 0.75) and has a diameter of 50 cells, while the slot has a width of 6 cells and a height of 30 cells. The Courant number is set to 0.25 and the time step is such that one rotation corresponds to 2524 time steps. The results of the test are given in Figure 4.3.
CHAPTER 4. VALIDATION AND TEST RESULTS

Once again, there is good qualitative agreement between the initial and final results. The circular shape is well preserved and the slot shows rounding only at its highest point, even after a large number of time steps. The clockwise rotation of the field causes a bias in the slot deformation in the direction of rotation. The behavior presented in Figure 4.3b is in excellent agreement with the solution computed by Ubbink [48], which shows the same degree
of degradation in the slot profile, in addition to the rotational bias. The computed solution also compares well to those shown in [38]. The solution has a sharp interface resolution, and there is an absence of "Flotsam and Jetsam" that is common to the SLIC and Hirt-Nichols VOF methods, as shown in [38].

Table 4.2 gives error values for our CICSAM implementation in comparison to those of the previously discussed methods. Once again, our implementation gives values better than those in [38] (with the exception of Youngs method), but underperforms when compared to the original implementation of CICSAM. This under performance is not an error within the method, but again most likely due to the fact that our implementation and actual solution method used for the solution of the volume fraction equation are different than the type used by Ubbink in [48, 49], and as discussed in the circular translation test, the CICSAM method will try and convert our jagged profile to a smooth circle. As seen in Figure 4.3b, CICSAM can maintain some of the jagged features, but still smooths out the majority of the profile, as expected. The results given in Table 4.2 and Figure 4.3b are in very good agreement with the literature, showing that our implementation deals with the rotational tests quite well.

<table>
<thead>
<tr>
<th></th>
<th>SLIC</th>
<th>Hirt-Nichols</th>
<th>FCT-VOF</th>
<th>Youngs</th>
<th>Ubbink</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split Circle</td>
<td>8.38E-02</td>
<td>9.62E-02</td>
<td>3.29E-02</td>
<td>1.09E-02</td>
<td>1.62E-02</td>
<td>2.81E-02</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of error values for rotation tests

4.1.3 Shear Flow

As discussed in [38, 48], translational and rotational tests are insufficient to accurately judge the performance of an interface tracking scheme, as the initial profile is not subjected to deformation due to shear. In most realistic flows, the interface shape has a considerable deformation, so the performance of the solution method needs to be judged in these cases. An appropriate shearing flow is given in [38, 48] as

\[ \mathbf{u} = < \sin x \cos y, -\cos x \sin y > \]  

(4.2)

Here, a uniform structured mesh of dimensions \( x, y \in [0, \pi] \) is used at a resolution of 100 \( \times \) 100 cells. Initially, a circle of volume fraction with radius 0.2\( \pi \) is placed on the mesh with its center located at (0.5\( \pi \), 0.2(1+\( \pi \))). The initial profile and velocity field are shown in Figure 4.4. The test is run at a Courant number of 0.25 for a given number of steps, \( N \), and then the resulting solution is integrated backwards in time for the same number of steps by
reversing the direction of the velocity field. Theoretically, under advection, the initial profile is returned exactly, and therefore the error in the computed solution is easily calculated. The results of this test are given in Figure 4.5 for the cases of 500 and 1000 time steps.

![Figure 4.4: Initial profile and velocity field for shear test](image)

For an increasing number of shearing steps, there should be more difficulty in returning the initial profile. In the case of 500 forward and reversed steps, Figure 4.5b shows our code does a good job of returning the initial circular profile, but there is some smearing of the interface and a slight deformation in shape. In the case of 1000 time steps, a much higher degree of shearing has taken place and the circular profile is not returned as well as in the previous case. There is the same degree of interface smearing, but there is a larger amount of deformation in the original profile. However, the result given in Figure 4.5d is completely consistent with the profiles obtained by Ubbink [48]. Our final profile exhibits the same amount and type of deformation as reported in [48]. It is worth noting that in Figure 4.5c, the tail of the profile shows volume fraction values that are yellow in colour. These represent regions where the grid density is too coarse to resolve the interface and both sides of the tail interface fall within a given cell. This is a difficult task for an interface capturing method and the volume fraction is assigned a value between zero and one to signify that this has occurred. This is discussed further in Section 4.3.
CHAPTER 4. VALIDATION AND TEST RESULTS

Table 4.3: Comparison of error values for shear tests

<table>
<thead>
<tr>
<th>N</th>
<th>SLIC</th>
<th>Hirt-Nichols</th>
<th>FCT-VOF</th>
<th>Youngs</th>
<th>Ubbink</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>2.72E-02</td>
<td>3.24E-02</td>
<td>1.94E-02</td>
<td>2.61E-03</td>
<td>1.63E-02</td>
<td>3.31E-02</td>
</tr>
<tr>
<td>500</td>
<td>3.30E-02</td>
<td>4.00E-02</td>
<td>2.35E-02</td>
<td>5.12E-03</td>
<td>2.09E-02</td>
<td>3.26E-02</td>
</tr>
<tr>
<td>1000</td>
<td>4.59E-02</td>
<td>6.60E-02</td>
<td>3.14E-02</td>
<td>8.60E-03</td>
<td>2.90E-02</td>
<td>3.95E-02</td>
</tr>
</tbody>
</table>

The errors for the cases of $N = 250, 500, 1000$ are reported in Table 4.3. For $N = 250$, our implementation gives a larger error value than for the other cases. This is once again due to difference in the circular volume fraction profile returned as result of the computation and the one used as the initial condition, as discussed earlier. Hence all the “current” error values reported in Table 4.3 have the same discrepancy and lie in the same range. It would
CHAPTER 4. VALIDATION AND TEST RESULTS

seem this error dominates any error due to deformation at \( N = 250 \) and \( N = 500 \). However, as the number of time steps increases to \( N = 500 \), our method essentially outperforms the methods discussed in [38], except for Youngs’ method. For \( N = 1000 \), our method gives very good quantitative results in comparison to the other methods. The results are also somewhat misleading in that the visual data given in [38] (shown in Figure 4.6) clearly show that all of the methods except for Youngs’ give a great deal of “Flotsam and Jetsam”, where as our solutions do not exhibit this behavior. The results are once again consistent with those obtained by Ubbink, with differences in numerical values due to the issues discussed above. Based upon the results presented and their comparison to those available in the literature, we safely assume that our implementation of CICSAM is valid and correct.

![Figure 4.6: Volume fraction profiles for 1000 forward and reversed shearing steps using different methods (From Rudman [38])](image)

4.2 Surface Tension Test Cases

With our interface tracking method tested and in place, the implementation of the surface tension modifications can be checked with further test cases. These tests make use of a fully coupled interface tracking computation through the solution of the mass and momentum equations in conjunction with the volume fraction equation.
4.2.1 Circular Drop

The standard surface tension test case is the prediction of the pressure change across a circular drop of a heavier fluid placed inside a lighter fluid with different physical properties. According to Laplace’s Law [3, 49, 51], for a two-dimensional bubble there is a pressure drop given by

$$\Delta P = \frac{\sigma}{R}$$

In this case, the larger pressure is on the concave side (within the drop). Using this result, we test the results obtained for the computed pressure difference. Following the example set by Brackbill et al [3], we use uniform meshes of dimensions $x,y \in [6cm, 6cm]$, with resolutions of both $30 \times 30$ and $60 \times 60$ cells. We place a bubble of radius 2 cm at the center of the mesh. The inner drop fluid has a density of $\rho_{in} = 1000\text{kg/m}^3$, while the outer density is half this value, $\rho_{out} = 500\text{kg/m}^3$, and the surface tension coefficient is prescribed a value of $\sigma = 0.02361\text{N/m}$. Through numerical experimentation, it is determined that the ratio of outer to inner viscosity values is unimportant to the result of the calculation. Therefore, in anticipation of future test cases, the inner viscosity is assigned the value of $\mu_{in} = 1.0 \times 10^{-3}\text{kg/m \cdot s}$, corresponding to that of water, while the outer viscosity is assigned a value of $\mu_{out} = 1.8 \times 10^{-5}\text{kg/m \cdot s}$, corresponding to the viscosity of air at room temperature.

The grid has boundary conditions such that the pressure is held at a constant value of $P = 0.0\text{Pa}$. For these parameters, the expected pressure difference is $\Delta P = 1.1805\text{Pa}$. This test case is run using both the Laplacian filter based CSF and convolution based (hereafter known as “Hybrid”) methods, as discussed in Section 3.5. We use an initial profile that is as close to circular as possible (the “jagged” circle, as in Figure 4.4), and the solution is run until a “steady-state” is reached (where there is a lack of significant change in the calculated pressure). The Courant number is kept at approximately 0.15 for each test run. The final circular profile and pressure distribution are shown in Figure 4.7 for a mesh resolution of 60 $\times$ 60 cells using the CSF method.

According to theory, the final profile is a circle with a zero velocity field. The computed shape is indeed circular, as expected. The key feature of Figure 4.7 is the presence of a non-zero velocity field around the bubble, dubbed “parasite currents” by Lafaurie et al [19]. This velocity field is always present and is an inherent feature of this type of continuum surface tension formulation [3, 19, 49, 51]. As discussed in [19, 49], the parasite currents depend on the ratio of the surface tension coefficient to the viscosity. Hence for specific physical problems where these constants are fixed, there is no control over the magnitude of the currents. Therefore, no further attempt is made to control them through modification.
CHAPTER 4. VALIDATION AND TEST RESULTS

Figure 4.7: Results of surface tension test case using CSF method
CHAPTER 4. VALIDATION AND TEST RESULTS

of these parameters. It is important to note that the parasite currents are not a steady state
phenomenon, but grow in time due to the accumulation of numerical error within the surface
tension scheme. Hence a rough qualitative estimate of the quality of a given surface tension
method is through the size of its parasite currents. The smaller this velocity field, the better
the method, as the theoretical result is more closely replicated. There is excellent qualitative
agreement between Figure 4.7 and similar results presented in [49, 51]. The width of the
interface and the pressure distribution obtained are very similar to the results mentioned
above and exhibit the correct behavior.

Quantitative estimates of the solution quality are made by examining various quantities
as in [3]. The computed mean pressure of the drop interior is found by calculating

\[
\langle P \rangle = \frac{1}{Nd} \sum_{i,j,k=1}^{Nd} P_{i,j}
\]

where \( Nd \) represents all cells in which the volume fraction \( \alpha \geq 0.99 \). For an initial pressure
of 0.0 Pa, the theoretical interior drop pressure is \( P_{\text{drop}} = 1.1805 \text{Pa} \), as discussed above. The
ratio of the computed mean pressure to the theoretical mean pressure, \( \langle P \rangle / P_{\text{drop}} \) is also
used as an estimate of the error, as a perfect result would return a ratio of 1.0. In addition,
the \( L_2 \) norm is used to estimate error, and is computed as

\[
L_2 = \left( \frac{\sum_{i,j,k=1}^{Nd} (P_{i,j,k} - P_{\text{drop}})^2}{Nd P_{\text{drop}}^2} \right)^{1/2}
\]

Finally, the curvature of a circle is simply given as

\[
\kappa = \frac{1}{R}
\]

where \( R \) is the radius of the circle. For this problem, \( R = 0.02 \text{m} \) and the curvature is
\( \kappa = 50 \text{m}^{-1} \). The computed mean curvature is given as [3]

\[
\langle \kappa \rangle = \frac{\langle P \rangle}{\sigma}
\]

The errors in the computed results and those given in [3] (listed as BKZ) for their MAC
scheme are listed in Table 4.4 for grid densities of 30 and 60 cells.

The Hybrid method does a much better job of matching the errors produced in [3] than
our implementation of the CSF method at the 30 \times 30 mesh resolution. However, this is a
CHAPTER 4. VALIDATION AND TEST RESULTS

Table 4.4: Error values for surface tension tests

<table>
<thead>
<tr>
<th></th>
<th>BKZ 30</th>
<th>BKZ 60</th>
<th>CSF 30</th>
<th>CSF 60</th>
<th>Hybrid 30</th>
<th>Hybrid 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt; P &gt; / P_{drop}$</td>
<td>1.153</td>
<td>1.043</td>
<td>0.863</td>
<td>0.906</td>
<td>1.145</td>
<td>1.249</td>
</tr>
<tr>
<td>L2 norm</td>
<td>1.65E-01</td>
<td>5.14E-02</td>
<td>1.37E-01</td>
<td>1.01E-01</td>
<td>1.44E-01</td>
<td>2.06E-01</td>
</tr>
<tr>
<td>Curvature</td>
<td>57.65</td>
<td>52.15</td>
<td>43.15</td>
<td>45.28</td>
<td>57.22</td>
<td>62.46</td>
</tr>
</tbody>
</table>

misleading result due to the fact that the Hybrid method performs so poorly compared to the CSF method at a $60 \times 60$ resolution. The CSF method gives results that consistently under predict (10-15%) when compared to theory, while our Hybrid formulation tends to overpredict solution values by a much greater amount. As is the case with the volume fraction test cases, the method used to compute the solutions in [3] is completely different than the method used in the $CMGFD$ code, and a different smoothing technique is used for the volume fraction field. We therefore expect differences in the results cited when compared with the current values. Since the computed results for CSF fall within 10-15% of theory, we can assume these solutions provide reasonably accurate estimation of surface tension effects. The computed profiles and numerical results for the CSF method presented above suggest a successful solution to this test case, while the current implementation of the Hybrid method does an inadequate job of simulating the correct drop behavior. This is a surprising result, as the Hybrid method is generally accepted as being a more accurate scheme than the original CSF formulation [4, 51]. Due to the results presented in Table 4.4, only the CSF method is used to obtain further surface tension results in this study.

We also examine the CSF method by investigating the effects of changing the number of passes of the Laplacian filter used to smooth the volume fraction field. The results of Table 4.4 use a two-pass filter for all calculations. Increasing the number of passes gives a smoother field for curvature calculations, but it is also less physically accurate, as the volume fraction values are spread further across the transition region. The above results are repeated using four passes of the filter, and presented in Table 4.5

Table 4.5: Error values for surface tension tests using different laplacian filter passes

<table>
<thead>
<tr>
<th></th>
<th>BKZ 30</th>
<th>BKZ 60</th>
<th>CSF 30-2</th>
<th>CSF 60-2</th>
<th>CSF 30-4</th>
<th>CSF 60-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt; P &gt; / P_{drop}$</td>
<td>1.153</td>
<td>1.043</td>
<td>0.863</td>
<td>0.906</td>
<td>0.918</td>
<td>0.931</td>
</tr>
<tr>
<td>L2 norm</td>
<td>1.65E-01</td>
<td>5.14E-02</td>
<td>1.37E-01</td>
<td>1.01E-01</td>
<td>1.14E-01</td>
<td>9.06E-02</td>
</tr>
<tr>
<td>Curvature</td>
<td>57.65</td>
<td>52.15</td>
<td>43.15</td>
<td>45.28</td>
<td>45.91</td>
<td>46.58</td>
</tr>
</tbody>
</table>
**CHAPTER 4. VALIDATION AND TEST RESULTS**

For a fixed cartesian mesh, an increased number of passes gives a better match with theoretical results, as there is closer agreement of the calculated quantities. This suggests better answers are achieved if a higher number of passes is used. However, this idea is examined further by calculating results using a circular mesh instead of a cartesian grid. This will allow the initial circular shape to be correctly produced (instead of the "jagged" initial profile) and give better insight into the results. An example of the initial profiles using the cartesian and circular grids is given in Figure 4.8. The test is repeated using the two and four pass filters on a $30 \times 30$ circular mesh. The results are listed in Table 4.6.

![Initial distributions using different meshes, 30 x 30 resolution](image)

(a) Cartesian mesh  
(b) Circular mesh

**Figure 4.8: Initial distributions using different meshes, 30 × 30 resolution**

<table>
<thead>
<tr>
<th></th>
<th>BKZ 30</th>
<th>BKZ 60</th>
<th>CSF 30-2</th>
<th>CSF 30-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{P}{P_{\text{drop}}} )</td>
<td>1.153</td>
<td>1.043</td>
<td>0.981</td>
<td>0.924</td>
</tr>
<tr>
<td>L2 norm</td>
<td>1.65E-01</td>
<td>5.14E-02</td>
<td>1.01E-01</td>
<td>1.26E-01</td>
</tr>
<tr>
<td>Curvature</td>
<td>57.65</td>
<td>52.15</td>
<td>49.03</td>
<td>46.24</td>
</tr>
</tbody>
</table>

**Table 4.6: Error values for surface tension tests using circular meshes**

From Table 4.6, it is evident that a higher number of passes on circular meshes leads to poorer results than with a lower number on the same mesh. The results of the two pass test are in excellent agreement with theory, suggesting that this is the optimal number of passes to use. Therefore, the improved results obtained with more passes on the cartesian mesh are considered an artifact of the method and its behavior on the that grid. Due to the results on the circular mesh, the two pass filter is used for the remainder of the study.
CHAPTER 4. VALIDATION AND TEST RESULTS

It is of particular interest to simulate high-density ratio flows (i.e., water and air). Therefore, we extend the scope of this test case by increasing the density ratio between the fluids in an attempt to determine the accuracy of the results produced in these cases. The above case is repeated for increasing density ratios using the CSF method with a $60 \times 60$ rectangular mesh. The error values are listed in Table 4.7.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\frac{P^0}{P_{drop}}$</th>
<th>L2 norm</th>
<th>Curvature</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.9946</td>
<td>8.11E-02</td>
<td>49.76</td>
</tr>
<tr>
<td>10</td>
<td>1.0191</td>
<td>1.07E-01</td>
<td>50.95</td>
</tr>
<tr>
<td>50</td>
<td>1.0674</td>
<td>4.83E-01</td>
<td>53.37</td>
</tr>
<tr>
<td>100</td>
<td>1.1905</td>
<td>9.52E-01</td>
<td>59.53</td>
</tr>
</tbody>
</table>

Table 4.7: Error values for surface tension tests with varying density ratios, rectangular mesh

Beyond a density ratio of 50, the quality of the results decrease significantly. This degradation with increasing density ratios is discussed throughout the literature [17, 39] and is a common (and still mostly unresolved) source of error for most continuum based surface tension methods, although various attempts are being made to alleviate this and other issues relating to these methods [18]. The discrepancies between computed and theoretical results given here tend to have a maximum error of ±10% (up to a density ratio of 50), which is acceptable as continuum surface methods have not yet been perfected to a general consensus within the literature. Accurate prediction of very high density ratios (> 100) cannot be simulated as well as would be desired on a rectangular mesh at these given resolutions.

The same tests are also computed on a $30 \times 30$ circular mesh for purposes of comparison. The values are listed in Table 4.8

<table>
<thead>
<tr>
<th>Ratio</th>
<th>$\frac{P^0}{P_{drop}}$</th>
<th>L2 norm</th>
<th>Curvature</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.991</td>
<td>1.12E-01</td>
<td>49.54</td>
</tr>
<tr>
<td>10</td>
<td>0.995</td>
<td>1.27E-01</td>
<td>49.77</td>
</tr>
<tr>
<td>50</td>
<td>0.999</td>
<td>1.36E-01</td>
<td>49.98</td>
</tr>
<tr>
<td>100</td>
<td>1.000</td>
<td>1.37E-01</td>
<td>50.01</td>
</tr>
<tr>
<td>1000</td>
<td>1.000</td>
<td>1.38E-01</td>
<td>50.04</td>
</tr>
</tbody>
</table>

Table 4.8: Error values for surface tension tests with varying density ratios, circular mesh

The above values show that the much coarser circular grid can produce results that match exactly with theoretical results, at any of the given density ratios. The error norms do slightly increase for higher density ratios, but the rise is small and the values clearly match with theory. The circular grid does not suffer from the same problems with density ratios as does the rectangular grid.
CHAPTER 4. VALIDATION AND TEST RESULTS

The surface tension modifications to CMGFD obviously reproduce the correct surface tension values for the given test case. The final comparisons show the great sensitivity of the results to the type of mesh used (rectangular vs. circular) and the mesh resolution (for rectangular meshes). A very fine grid would be required for an accurate reproduction of the theoretical results on a cartesian mesh. This is extremely prohibitive for the current implicit implementation, as the run times would be ridiculous and not useful for engineering estimates. The results for the circular meshes show that the CMGFD code produces the correct values for surface tension flows.

4.2.2 Non Circular Drop

The action of the surface tension force is to convert any non-circular shape into a circle, in order to minimize its surface energy. Therefore, using a non-circular initial profile always results in a final circular shape for any given test problem. An extreme case exemplifying this behavior is described in Ubbink [49] and Brackbill et al [3], whereby a two-dimensional initially square profile is acted upon by surface tension, converting it to a circle. Corresponding to [3], we use a grid of size $x, y \in [7.5\,\text{cm}, 7.5\,\text{cm}]$ with a grid resolution of 30 x 30 cells, with constant pressure boundary conditions, held at 0.0Pa. The density of both the inner and outer fluids is set at $\rho = 797.88\,\text{kg/m}^3$, with an artificial surface tension coefficient of $\sigma = 0.02361\,\text{N/m}$. The square is placed in the center of the mesh, and has a length of 18 cells. There are no quantitative results listed in [3], but a series of figures showing the behavior of the drop at various times is given. These are reproduced in Figure 4.10 at the time values cited in [3].

As discussed in [3, 49], the initially square volume fraction profile is acted upon solely by the surface tension force. The corners represent regions of high curvature, and hence the surface tension force acts inwards, as shown in Figure 4.10a. This causes the square profile to deform into a circle, as chronicled in Figures 4.10b-f. However, there is sufficient numerical error within the free surface method to drive the relaxation of the drop past the circular shape, leading to the onset of oscillatory motion. The oscillations repeat over many cycles until there has been sufficient numerical dissipation to cause the surface tension force to subside. The final shape is circular, as expected, and the curvature is uniform, meaning there is no motivating force with which to cause any further deformation of the bubble. The qualitative behavior of the computed results is logical and matches the of the above listed works. This further verifies the ability of our surface tension modifications to function with satisfactory accuracy.
CHAPTER 4. VALIDATION AND TEST RESULTS

Figure 4.9: Relaxation of Square Drop

(a) $t = 0.0s$

(b) $t = 0.05s$

(c) $t = 0.10s$

(d) $t = 0.20s$

(e) $t = 1.00s$

(f) $t = 2.00s$
4.3 Rayleigh-Taylor Instability

Another test case for two-fluid systems is that of Rayleigh-Taylor instability. A particular version of this problem has a column of fluid resting upon another column of fluid with smaller density, bounded by rigid vertical walls and subject to gravitational and surface tension forces. If at the fluid interface a small perturbation is applied, the disturbance should grow exponentially in time, and the unstable configuration will collapse. A mixing of the fluids will take place, with the denser fluid displacing the lighter fluid and occupying the space below it. This phenomenon is well known [2, 7], but unfortunately no analytical solutions exist for the transient fluid behavior. There are several results describing the stability of the system subject to different types of perturbation [7]. However, we are not interested in obtaining results for the growth rate of the disturbances in this study, but instead wish simply to test the ability of our code to qualitatively simulate this interesting problem. No attempt is made to quantitatively compare the solutions to any prior known results.

We investigate the dynamics of this problem by using a computational domain with dimensions $x, y \in [1m, 3m]$, and a grid density of $30 \times 90$ cells. The fluids used in this simulations are both viscous, and have a viscosity value of $\mu = 0.01kg/m \cdot s$. The upper fluid is given a density of $\rho = 2kg/m^3$, while the lower fluid has a density of $\rho = 1kg/m^3$. The gravitational acceleration is set at $g = 9.8m/s^2$, and the surface tension coefficient is assigned a value of $\sigma = 0.01N/m$. The fluids are initially at rest, and the interface is given an initial velocity perturbation downwards to initiate the instability growth. The maximum Courant number during the calculation is 0.3.

The result of the simulation is shown for different times in Figure 4.10. The results show the expected motion of the denser fluid acting to displace the lighter fluid, and occupy the region below it. In Figures 4.10 a-c, the separation of the two fluids is well defined, as indicated by the regions of red and blue. However, in Figure 4.10c, we see the beginnings of a potential trouble spot in the “tail” region of the falling spike of dense fluid. As the fluid falls, the tail becomes thinner. In Figure 4.10c, the tail is still sufficiently thick that the grid is resolved well enough to capture the interface sharply, as both vertical interfaces of the tail fall within different cells. In Figure 4.10d, the tail has become so thin that both interfaces of the tail now fall within a single cell, and the volume fraction method has difficulties in coping with the situation. CICSAM assigns a composite volume fraction value lower than unity to these cells in an attempt to convey the information that the cell contains two interfaces. Physically, the interfaces are still present, but numerically (due to the coarse mesh resolution used), the method cannot interpret the situation correctly. This breakdown
also occurs in the upper corners of Figure 4.10d, where the denser fluid begins to fold over itself, and the resolution is again too coarse to capture the interfaces properly. Once these regions have developed within the flow and intermediate volume fraction values are assigned in non-interfacial regions, these "approximated" parcels of fluid are advected along by the numerical method and are propagated throughout the flow, causing the transition region of the fluid in Figures 4.10 e,f to grow. Since the coarse mesh encourages this behavior, it eventually dominates the solution profiles. As the calculation proceeds forwards in time, the denser fluid proceeds downwards to occupy the lower region of the box. Letting the simulation continue, we find that the volume of dense fluid is regained at the bottom of the box, but due to the artificial approximations made, the correct mass is not recovered. This means the method cannot reconstitute the fluid back to its original state once the approximations have been made.

Based on previous numerical simulations of inviscid Rayleigh-Taylor problems ([3, 38, 52]), the current results do exhibit the correct type of qualitative behavior associated with this class of problem, especially during the earlier stages of computation prior to and including the spike of heavier fluid falling downwards. The results show that there is a strong relationship between solution quality and the resolution of mesh used. If the mesh is too coarse, regions where more than one interface may occur in a given cell manifest themselves earlier than with a finely resolved mesh, causing a degradation in the simulation results. Depending on the problem being solved and how complex the dynamics involved are, a very fine mesh is required, particularly near the interface. The calculation is not repeated for a finer grid, as the run times would increase by too large a factor to make the solution at the higher resolution worth while. It is important to note that all interface capturing schemes suffer from this malady, and this behavior is an inherent trait of the method and should not be viewed as a reason to abandon the method altogether; the remedy is to increase the mesh resolution. The results presented here show a qualitatively successful result with regards to the simulation of a Rayleigh-Taylor instability, and more importantly show that the modified CMGFD code can handle a "real" problem that includes a deforming surface and the presence of relatively moderate gravitational and surface tension forces.
Figure 4.10: Rayleigh-Taylor instability at various times
4.4 Sloshing

Another useful test of the free surface capabilities of a computer program is the ability to reproduce the sloshing of fluid within a container with rigid walls. There are analytical and computational results presented for several inviscid cases of sloshing [49], but none reported for viscous sloshing. We are again interested in the qualitative ability of our code to produce results for this case. In particular, we wish to judge the ability of the CMGFD code to simulate the behavior of fluids in the presence of relatively large gravitational forces due to high density ratios between fluids. The simulation involves two fluids, air and water, in a box with an open top. The mesh used is non-uniform and non-orthogonal in order to define a highly resolved, unstable initial fluid configuration in the presence of gravity. The mesh is shown in Figure 4.11 and has a resolution of $32 \times 36$ cells, with the cell concentration higher near the initial location of the interface. The density of the water is $\rho_w = 998 \text{ kg/m}^3$, while the air density is $\rho_a = 1.2 \text{ kg/m}^3$. The water viscosity is set at $\mu_w = 1 \times 10^{-3} \text{ kg/m} \cdot \text{s}$ and the air at $\mu_a = 1.2 \text{ kg/m} \cdot \text{s}$. The surface tension coefficient has a value of $\sigma = 0.073 \text{ N} \cdot \text{m}$ and the acceleration due to gravity is $g = 9.8 \text{ m/s}^2$. The initial configuration is that of a gentle slant, and is shown along with results of the simulation at various times in Figure 4.12.

![Figure 4.11: Grid used for sloshing tests](image)

The simulation does not proceed to give the expected oscillations associated with sloshing. Instead, there is a divergence of the solution at one of the corners where the fluid interface resides. Numerical experimentation shows that this is due to the high density ratio ($\approx 830$) in air-water systems in the presence of large gravitational and surface tension body...
forces in the momentum equations. Lowering the time step or refining the mesh causes no improvement in the solution result, and simply extends the run time until the divergence occurs. The calculation is repeated with the air density set to \( \rho_a = 2.4 \text{kg/m}^3 \), (double that of normal air) so that the density ratio of the fluids is \( \approx 415 \). All other physical parameters are kept constant, and the solution is repeated. The results are shown in Figure 4.13. The lower density ratio permits calculation of the sloshing case for this mesh resolution and a relatively moderate time step of \( \Delta t = 1 \times 10^{-3} \text{sec} \). The expected sloshing behavior is present, with gentle oscillations continuing at low amplitudes for many periods, until the entire system comes to rest. These results, coupled with results performed on various other test cases not mentioned here, suggest that our free-surface modification to the \textit{CMGFD} code is very sensitive to the density ratio of the fluids when high density ratios and gravitational body forces are used. In specific for the case of air-water systems, the density of the air may need to be increased by a factor of 2-4 times, giving density ratios ranging from approximately 200-415, in order to obtain a successful result. These modified air density values are of the same order of magnitude as the original density and are valid approximations to make in the interest of computing converged solutions. For these lower density ratios, the behavior of the air vectors is less erratic than for the proper water-air cases, as the gas is less sensitive to changes in flow, both physically and numerically. Since the water properties are kept at their standard values and only the gas density is varied, then to an acceptable modification, we assume that our code can deal with high density ratio body force flows as long as the ratio is kept below 500, which is acceptable when trying to simulate air-water flow behavior and properties.
Figure 4.13: Sloshing in tank at various times
4.5 Chapter Summary

This chapter deals with the results computed by the modified CMGFD code for a number of standard test cases. The code reproduces values that are in agreement with results presented in the literature for various volume fraction tests. The surface tension capabilities of the code are also correctly implemented. The program can simulate a variety of actual flow fields, ranging from low to relatively high density ratios. The quality of result is greatly dependent on the mesh resolution used, with higher grid densities, particularly near the interface, generating improved results as expected. For higher density ratios (>500) in the presence of body forces, there is a large sensitivity on the density ratio of the fluids which can cause unnatural deformations in the surface profiles or divergence in the solution itself. A remedy for improved convergence in these cases is to increase the density of the gas phase by a factor of two to four in order to give a more stable result.
Chapter 5

Planar Jet Simulation

The CMGFD code is modified to deal with two phase flows and can account for surface tension effects. The program is now used to simulate various planar jet profiles, in an attempt to achieve the goal of simulation of experimentally observed surface waves on a planar jet.

5.1 2-D Jet in the Absence of Surface Tension and Gravity

While the CMGFD code is fully capable of simulating three-dimensional flows, the first case tested involving planar jets is that of a steady two-dimensional viscous jet in the absence of surface tension and gravity. This is a further test demonstrating the ability of the modified CMGFD code to couple the volume fraction and momentum equations through the solution of a realistic flow. This particular case possesses an analytic solution, and therefore the accuracy of the computed result can be determined. Using the boundary layer equations and an asymptotic analysis, Tillet [46] calculates the analytical solution for a laminar, steady state jet of viscous liquid exiting a channel with a fully developed parabolic velocity profile into vacuum. The free surface profile is determined to have the following form

\[ \zeta(x) = 0.70798 \left( \frac{h^2 x}{Re} \right)^{1/3} \]  \hspace{1cm} (5.1)

Here, \( x \) is the downstream location relative to the jet exit plane, and \( Re \) is the Reynolds number of the flow exiting the channel, based on the average velocity of the flow in the channel.
CHAPTER 5. PLANAR JET SIMULATION

channel, $U_m$, and the channel height, $h$

$$Re = \frac{\rho U_m h}{\mu}$$  \hspace{1cm} (5.2)

where $\rho, \mu$ are the density and dynamic viscosity of the fluid, respectively. The profile calculated in (5.1) is valid for downstream distances $x$ such that $x^{-1} \sim O(\frac{1}{Re})$. The results presented by Tillet are nearly identical (within 2\% to those based on the work of Goren [11]. Goren performed experimental studies relating to planar jets [12], and it is discovered that the experimental result differs from the result presented in (5.1) by approximately a factor of $1/2$. While the source of the discrepancy is unknown, the additional factor of $1/2$ applied to (5.1) gives results that are in good agreement with experiment [12]. Hence, the correction factor is used in this study as a basis of comparison for the computed and exact results. The modified surface location becomes

$$\zeta(x) = 0.35399 \left(\frac{h^2 x}{Re}\right)^{1/3}$$  \hspace{1cm} (5.3)

This profile is used to obtain a direct quantitative estimate of the error involved in the solution of this problem. To do so, we compare the computed solution to the exact through the square norm, $L_2$, defined as

$$L_2 = \left(\frac{\sum_{i,j} (\zeta^e - \zeta^c)^2}{I_{max} J_{max}}\right)^{1/2}$$  \hspace{1cm} (5.4)

where $\zeta^e$ is the exact solution, as given by (5.3), $\zeta^c$ is the computed solution, and $I_{max}, J_{max}$ are the maximum number of cells in each coordinate directions of the mesh.

Before computing the solution, we note that the current VOF modification of the CMGFD code cannot artificially set the density (or viscosity) of one of the fluids to zero, and hence cannot truly simulate the analytic result in accordance with the assumptions used in the derivation. However, we assume that for a high enough density ratio between the two fluids, the shear of the outer fluid will have minimal effect on the jet solution predicted by theory. Therefore, we solve the problem using two distinct fluids in the hopes of obtaining a suitable result. The problem is initially approached as a jet of water entering stationary air at atmospheric pressure. The properties of water used are $\rho_w = 998 kg/m^3, \mu_w = 1.0 \times 10^{-3} kg/m \cdot s$, while those of air are $\rho_a = 1.2 kg/m^3, \mu_a = 1.8 \times 10^{-5} kg/m \cdot s$. 

5.1.1 Boundary Conditions

The choice of numerical boundary conditions is of great importance to the results of the simulation. A schematic showing the domain and boundary conditions is given in Figure 5.1. There is an inherent vertical symmetry in this problem, so the jet is computed for the half-region above the centerline of the channel, i.e., for \( y > \frac{h}{2} \). Therefore, the south boundary of the domain is taken to be the jet centerline and assigned a "symmetry" or "free-slip" condition on the velocity and volume fraction fields, namely \( \frac{dv}{dy} = 0 \) and \( \frac{da}{dy} = 0 \).

According to Tillet [46], the liquid jet exits into open vacuum. We cannot directly simulate this case, but instead use a domain filled with stationary air at atmospheric pressure. This corresponds to a grid with a constant pressure boundary condition, \( P = P_{atm} \). In order to compute steady flows with the CICSAM method, we must obey the Courant number limitation in each time step and march in time towards a steady-state solution. Therefore, the solution method is inherently transient. Solutions to any problem using this method cannot be computed with a steady-state solver (which would be the natural choice when computing any solution profile that does not vary in time). For this unsteady calculation, the constant pressure boundary condition prevents convergence of the numerical solution by drawing a large amount of air into the grid, causing the solution to diverge during the early stages of computation. As a result, numerous different boundary conditions are investigated, and the ones that provide the most suitable results are solid walls on the north, east, and west (excluding the inlet plane) boundaries of the domain. They are used at a suitable distance away from the jet and give a "no-slip" condition, namely \( u(x_w) = 0 \) (Dirichlet conditions), where \( x_w \) represents the location of the given wall. It is assumed that the influence of the walls is minimal if they are a suitable distance away from the jet. According to Ubbink [49], an appropriate boundary condition for the volume fraction at the walls is that of zero normal gradient (Neumann condition), \( \frac{da}{dn} = 0 \), where \( n \) represents the normal direction to the surface.

In order to permit fluid to leave the computational domain, an opening in the east boundary wall (for fluid flowing west to east) above the symmetry plane is provided by replacing the no-slip wall condition with a zero velocity gradient condition at the exit plane, \( \frac{du}{dn} = 0 \). The volume fraction also has a zero gradient condition, as discussed above.

The grid is extended in the direction west of the jet inlet (i.e., behind it), by adding a second, inter-connected domain segment. The east wall of this segment is considered an internal boundary of the mesh, as it is connected to the west boundary of the original segment. The west, south and north walls of the second segment are also assigned "wall" boundary conditions. A further zero velocity gradient opening is placed in the west wall of this second
CHAPTER 5. PLANAR JET SIMULATION

segment to allow air to enter the domain. Once again, many different boundary conditions are tested, and the above combination is the most suitable (and often times the only one producing a result at all!).

![Diagram of domain and boundary conditions used for jet test](image)

Figure 5.1: Schematic of domain and boundary conditions used for jet test

According to the solution given by Tillet [46], the water jet must have a fully developed parabolic velocity profile before entering the ambient air. At the inlet, the volume fraction boundary condition simply specifies that water is entering the domain, equivalent to a Dirichlet condition with $\alpha = 1$. The velocity profile is also known and in order to minimize computational time and effort, the fully developed flow profile is implemented as a boundary condition for the inlet of the numerical domain rather than being computed. For a viscous fluid in a channel extending from $y = 0$ to $y = h$, the well known parabolic velocity profile is

\[
u(y) = \frac{1}{2\mu} \frac{dP}{dx} (y^2 - yh)\]  

(5.5)

The maximum velocity, $U_c$, occurs at the center of the channel, $y = \frac{h}{2}$, so we solve for the pressure gradient, $\frac{dP}{dx}$, in terms of the centerline velocity, $U_c$

\[
\frac{dP}{dx} = -\frac{8\mu U_c}{h^2}\]  

(5.6)
Substitution of (5.6) into (5.5) yields

\[ u(y) = \frac{4U_c y}{h} \left( 1 - \frac{y}{h} \right) \]  

(5.7)

Using (5.7), we integrate across the channel height to determine the average velocity, \( U_m \)

\[ U_m = \frac{1}{h} \int_0^h \frac{4U_c y}{h} \left( 1 - \frac{y}{h} \right) = \frac{2}{3} U_c \]  

(5.8)

Therefore, we represent the centerline velocity in terms of the mean velocity, \( U_c = \frac{3}{2} U_m \), and the final form of the fully developed velocity profile used as a boundary condition is

\[ u(y) = \frac{6U_m y}{h} \left( 1 - \frac{y}{h} \right) \]  

(5.9)

This form is useful for most computations, as the mean velocity, \( U_m \), is more easily specified than either the centerline velocity, \( U_c \), or the pressure gradient, \( \frac{dp}{dx} \).

### 5.1.2 Numerical Solution

Using the above boundary conditions, we compute a numerical solution to this test problem. The grid used must maintain the resolution and integrity of the free surface. For a computed result where no analytical results are available, grid refinement studies are used to determine the proper mesh density for which grid independent results occur. In anticipation of the resulting surface profile given in (5.3), various non-uniform structured cells having a higher density near the expected surface location are used. The mesh density used in the solution results from a series of grid refinement studies performed during the process of investigating the current problem. As the analytical result is available, only the mesh used to obtain the final solution is presented and discussed. The physical dimensions of the computational domain used here correspond to suitable dimensions for a jet used by Söderberg [41] for experimental studies of planar jets, with a channel height of \( h = 1.1 \text{mm} \). The mesh is shown in Figure 5.2, while the dimensions are listed in Table 5.1. The grid used here is sufficiently fine to accurately capture the interface and model the expected solution behavior, while at the same time being coarse enough to minimize computer run times. The result of the computations is shown in Figures 5.3 and 5.4 for a jet of Reynolds number \( Re = 200 \) based on the channel height. To achieve steady-state profiles, the solutions are marched forward in time with a time step of \( \Delta t = 10^{-5} \text{sec} \) until there is negligible change in both the interface location and in the value of the square norm.


**Figure 5.2: Mesh used for jet simulation**

<table>
<thead>
<tr>
<th>side</th>
<th>dimensions</th>
<th>segment</th>
<th>cells</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>west</td>
<td>$x = 0, 0 \leq y \leq 4.18 \times 10^{-4}$</td>
<td>1</td>
<td>8</td>
<td>parabolic $u(y)$</td>
</tr>
<tr>
<td></td>
<td>$x = 0, 4.18 \times 10^{-4} \leq y \leq 5.5 \times 10^{-4}$</td>
<td>1</td>
<td>12</td>
<td>parabolic $u(y)$</td>
</tr>
<tr>
<td></td>
<td>$x = 0, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>13</td>
<td>internal</td>
</tr>
<tr>
<td>east</td>
<td>$x = 0.01, 0 \leq y \leq 5.5 \times 10^{-4}$</td>
<td>1</td>
<td>17</td>
<td>$\frac{\partial u}{\partial n} = 0$</td>
</tr>
<tr>
<td></td>
<td>$x = 0.01, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>16</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>north</td>
<td>$0 \leq x \leq 0.0011, y = 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>30</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>south</td>
<td>$0 \leq x \leq 0.0011, y = 0$</td>
<td>1</td>
<td>30</td>
<td>$\frac{\partial u}{\partial y} = 0$</td>
</tr>
<tr>
<td>west</td>
<td>$x = -0.0011, 5.5 \times 10^{-4} \leq y \leq 1.375 \times 10^{-3}$</td>
<td>2</td>
<td>10</td>
<td>$u = 0$</td>
</tr>
<tr>
<td></td>
<td>$x = -0.0011, 1.375 \times 10^{-3} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>3</td>
<td>$\frac{\partial u}{\partial n} = 0$</td>
</tr>
<tr>
<td>east</td>
<td>$x = -0.0011, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>13</td>
<td>internal</td>
</tr>
<tr>
<td>north</td>
<td>$-0.0011 \leq x \leq 0, y = 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>10</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>south</td>
<td>$-0.0011 \leq x \leq 0, y = 5.5 \times 10^{-4}$</td>
<td>2</td>
<td>10</td>
<td>$u = 0$</td>
</tr>
</tbody>
</table>

*Table 5.1: Dimensions and boundary conditions for jet simulations*
Figure 5.3: Computed solution for $Re = 200$

Figure 5.4: Computed and exact solutions for $Re = 200$
Clearly, the *CMGFD* code does a good job in simulating the expected jet profile. The interface shape is smooth and well computed and there is good qualitative agreement between the computed and exact solutions. Any discrepancies in shape are due to the level of resolution of the mesh used during solution and the fact that the exact solution is not a perfect prediction of the profile, but a very good estimate when compared with experimental results [12].

Solutions to this problem are computed using both the earlier described CICSAM implementation with the Crank-Nicolson flux averaging, as well as using the three-time level method (without the flux averaging). For the same time step size there are several differences in the solutions produced by the two methods. The Crank-Nicolson results have oscillations in the position of the surface profile where none are expected according to theory and experimental results [41]. This type of oscillatory behavior is commonly associated with the use of the Crank-Nicolson method [9]. Due to this oscillation, the Crank-Nicolson solutions need nearly twice the number of time steps than used for the three time level method for this transient behavior to diminish. However, the profile never settles into the expected steady state shape and the oscillations are always present. The final results between the methods are similar, but the three-time level method produces smoother overall profiles and gives slightly better agreement with theory, as indicated by the error values in Table 5.2. The Crank-Nicolson method also produces larger intermediate Courant number values during the inner iterations of the computation procedure (as discussed in Section 3.4), giving higher effective Courant number during the subsequent time steps of the simulation. Normally this leads to a more smeared interface with the CICSAM method. However, due to the greater degree of accuracy of the Crank-Nicolson method relative to the three time level method, the interface profiles computed by the latter are slightly more smeared than with the former. A comparison between the solutions produced by the methods is shown in Figure 5.5. The three-time level method is used to produce the solutions given Figures 5.3 and 5.4.

<table>
<thead>
<tr>
<th></th>
<th>Crank-Nicolson</th>
<th>Three-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_2$, $Re = 200$</td>
<td>4.106E-01</td>
<td>3.921E-01</td>
</tr>
</tbody>
</table>

Table 5.2: Error values for jet test, $Re = 200$

The errors in Table 5.2 show that there is slightly better accuracy in the final solution when using the three time level method. Although it gives a slightly more diffusive interface profile, it does produce the correct steady state shape, giving better agreement with the
analytic result. If our Crank-Nicolson solution was able to reach a true steady-state profile, it would have a more accurately resolved interface and hence would give a better result when compared to theory, as expected. However, the slight deformations in the surface, as seen from Figure 5.5b, cause it to underperform in comparison to the three time level method. The above values also imply that the flux-averaging procedure is not essential to the solution of the planar jet problem. Due to the similar results between both methods and the shorter run times associated with the three-time level method, it is used over the Crank-Nicolson based method for all further simulations in the remainder of the present section.

As noted earlier, the solutions presented above do not correspond to the exact mathematical situation described by Tillet due to the presence of two different viscous fluids. In light of the good agreement between the computed results with those in [46], we are curious to see how small the density ratio may be before the agreement with Tillet’s result starts to break down. The simulation is repeated for various fluid density ratios on the same mesh as the problems above. The qualitative profiles are given in Figure 5.6 and the error values are listed in Table 5.3.

<table>
<thead>
<tr>
<th>Density Ratio</th>
<th>$L_2, Re = 200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio=100</td>
<td>4.952E-01</td>
</tr>
<tr>
<td>Ratio=50</td>
<td>4.953E-01</td>
</tr>
</tbody>
</table>

Table 5.3: Error values for different density ratios, $Re = 200$

From Figure 5.6, the profiles remain similar to that of the air-water case for density ratios
Figure 5.6: Comparison of solutions with different density ratios, $Re = 200$

as low as 50, corresponding to an air density approximately 20 times larger than normal. As discussed in Sections 4.2 and 4.4, for high density ratio flows in the presence of gravity and surface tension, the density of the air may need to be increased for successful convergence of the numerical solution. The above results may therefore be useful when incorporating surface tension and gravity effects into jet results, as the computed profiles remain valid for increased gas densities. Based on the above, the solution is considered sufficiently accurate for relatively low density ratios.

When using any numerical method for engineering predictions, the computational run times needed to obtain the solution should be addressed. For this fairly simple two-phase flow problem on this relatively coarse mesh, the run time to steady-state is on the order of days using a computer with 1.5GHz processor. This is due to the nature of the jet. For this and any other steady flow problem, we march in time to compute the solution. In order to keep a tight interface resolution, the Courant number must be kept low, leading to very small time steps and an increased run time. Increasing the mesh resolution to further improve solution accuracy gives dramatically larger run times that prohibit such grid refinements, unless absolutely necessary. Therefore, there is a fine balance that must be struck between mesh density and run times when computing free surface flow problems using the standard iterative solution techniques that are found in the CMGFD code.

There is good agreement between the results produced with our modified code and those given by the analytical result. Our assumption that the presence of an external fluid will have a minimal effect on the jet seems to be valid, even for relatively low density ratios where the air density is much higher than its true physical value. The flux-averaging procedure
used in the CICSAM method is not needed to produce accurate results, and excluding the Crank-Nicolson time stepping gives solutions that are less prone to oscillation and take much less time to compute than with the three-time level method.

5.2 2-D Jet with Gravity

The next step in the simulation process is the inclusion of gravitational forces while still neglecting the effects of surface tension. To make quantitative studies of jets more amenable to mathematical analysis, the jet is aligned in the direction of gravity. In addition, any experimental study performed always aligns the jet in the direction of the field. The computations are performed using a water jet at a Reynolds number $Re = 200$ (based on channel thickness) exiting into air in the same direction as gravity, simulating a jet issuing vertically downwards. The grid from the previous test case is used with a time step of $\Delta t = 10^{-5}\text{sec}$. The gravitational acceleration is set to be $g = 9.8\text{m/s}^2$. All physical parameters of the air and water are kept as described in Section 5.1. The solution is given in Figure 5.7

![Figure 5.7: Computed profile for jet with gravity, $Re = 200$](image)

As in the previous case, the solution is run forward in time until there is no appreciable
change in the interface profile. There is a smooth behavior to the interface, similar to that of the previous case of a 2-D jet without gravity. The main difference between the solutions is that the presence of gravity causes a larger degree of contraction for the jet at the given Reynolds number, as seen from Figure 5.8. This is to be expected from mass continuity of viscous incompressible fluids; as the flow speeds up due to the acceleration of gravity, it must contract to satisfy mass conservation. Furthermore, an examination of the Froude number also supports this result. The Froude number represents the ratio of inertial forces to gravitational forces within a free surface flow and is defined as

\[ Fr = \frac{U^2}{gL} \]  

(5.10)

\( U \) is a characteristic velocity of the flow, taken to be the mean outlet velocity of the jet from the channel. For the case of a \( Re = 200 \) jet, \( U = 0.18m/s \). The magnitude of the gravitational acceleration is given by \( g = 9.8m/s^2 \), and \( L \) represents a characteristic length scale on which the problem is based. Here, the length of the computational domain, \( L = 1.1mm \), is taken as this length. For these parameters, the Froude number corresponding to a \( Re = 200 \) jet is \( Fr = 3.1 \). This value has an order of magnitude of unity, suggesting that there is
no clear dominance of either inertia or gravity on this length scale, so both effects should be visible in our simulation. This argument is supported by our computational results, as the larger contraction of the jet due to gravity is evident for this case. Unfortunately, there are no analytical results available for this test cases by which we can compare our results. However, on the basis of qualitative arguments the behavior of the jet is physically plausible and appears to be appropriate.

It is of worth to investigate the behavior of two-dimensional planar jets in the presence of gravity for higher Reynolds numbers. These cases are used later in this study for purposes of comparison with experimental results, and any insight into the physical effect of gravity on these higher Reynolds number profiles may be of use in interpreting future results. The two cases of interest are $Re = 1430$ and $Re = 1860$. These cases are simulated both with and without gravitational effects using the same mesh and physical parameters as above. A time step of $\Delta t = 10^{-6}$ sec. is used, and the simulations are again run until there is no appreciable change in the surface profile. The results are shown in Figure 5.9.

![Figure 5.9: Simulation of planar jets with gravity at different Reynolds numbers](image)

The simulation results are once again very plausible. There is a smooth interface profile that is physically realistic from a qualitative sense, as in the results of Figure 5.7. Comparing these solutions to those of jets at the same respective Reynolds numbers without gravitational effects gives the results of Figure 5.10.

It is clear that in these higher Reynolds numbers test cases, there is little or no further contraction of the jet on this length scale due to gravity. This behavior is expected from an examination of the Froude number in these cases. For $Re = 1430$, the outlet velocity is
CHAPTER 5. PLANAR JET SIMULATION

Figure 5.10: Comparison of planar jets with and without gravitational effects. Black=gravity, Red=without gravity

$U = 1.3\text{m/s}$, which gives a Froude number of $Fr = 157$. For $Re = 1860$, the outlet velocity is $U = 1.7\text{m/s}$, giving a Froude number of $Fr = 268$. In each case, the inertial effects clearly dominate over the gravitational forces, as indicated by the high Froude numbers. At the relatively low Reynolds number case of $Re = 200$, the Froude number is on the order of unity at the given length, so the contraction of the jet due to gravity is visible in our simulations. According to the Froude number, the effects of gravity at these higher Reynolds numbers are unimportant on our domain and no contraction should be visible. This is supported by the results of Figure 5.10.

Due to the small amount of contraction in the jets at higher Reynolds numbers, it is useful to compare the simulated results to the analytical results of Section 5.1 as a further check on our results at different Reynolds numbers. The error norms are listed in Table 5.4

<table>
<thead>
<tr>
<th></th>
<th>$Re = 1430$</th>
<th>$Re = 1430$ grav.</th>
<th>$Re = 1860$</th>
<th>$Re = 1860$ grav.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_2$ norm</td>
<td>1.904E-01</td>
<td>1.996E-01</td>
<td>2.021E-01</td>
<td>1.899E-01</td>
</tr>
</tbody>
</table>

Table 5.4: Error values for higher Reynolds number jets with gravity

There is much better agreement with the exact results for these higher Reynolds number cases than in the previous low Reynolds number jet. The results presented by Tillet [46] state that they are valid for “high Reynolds number” laminar jets, but give no specific values to use. As the phrase “high Reynolds number” is somewhat vague, we assume that the analytical results were intended for these higher Reynolds number cases and hence give better results. This is not to say that the results obtained for the $Re = 200$ case are invalid;
those results are in good agreement with the exact results. The current cases simply gives better agreement with the theory, further supporting the validity of our simulated results.

A final useful test is the ability of the code to produce reliable results for grids extended in the streamwise direction. In our previous examples, the computational domain has a length that is only a single exit channel diameter in magnitude. For future comparison of our simulations with experimental results, we need to extend this domain length by a much larger amount. For this reason, simulations are performed using a \( Re = 1430 \) jet with domain lengths of 1.1cm and 3.3cm, corresponding to a length of 10 and 30 diameters downstream, respectively. The simulations have a Froude number of \( Fr = 15.7 \) in the 10 diameter case, and \( Fr = 5.2 \) in the 30 diameter case, suggesting we should not see strong contraction effects due to gravity on either of these length scales, and that the results should be similar to the analytical results given by Tillet.

When extending the physical domain, we would ideally like to maintain the resolution of the current computational grid by adding an appropriate number of cells in the streamwise direction. However, since the cost of computation is already prohibitively high for the simulations with the previous meshes, increasing the number of cells by 10-30 times would be impractical. Hence, a compromise is made. The arrangement and relative sizes of the cells is kept constant as in Figure 5.2, while the physical length of the streamwise coordinate is increased by ten and thirty times respectively. This method does decrease the actual resolution of the solution obtained in a given region, but the same relative grid resolution is maintained for the entire simulation. A drawback to this approach is that the aspect ratio of the individual cells is also increased significantly. In the original grid, the aspect ratios of \( x \) to \( y \) cell lengths ranged from 0.25 to 3. The finest grids and highest aspect ratios occur near the interface, and are needed to keep sharp profiles. These values seem reasonable, and there seem to be no difficulties that arise from the use these cells. Indeed, the results of the current and last section show that there is very good agreement with available analytic results. We assume that this approach is valid when attempting to simulate behavior on a larger scale. We want to determine a suitable grid for which important physical phenomenon is resolved while ignoring the fine local detail offered through maintaining the previous local grid resolution and further adding cells.

For the extended meshes, the aspect ratios on the 10 and 30-times extended mesh range from 2.5-30 and 5-100, respectively. While these values are very high, these elongated cells give excellent agreement with analytical results, as shown in Figure 5.11. We liken this situation to a boundary layer calculation, where cells are stretched in the streamwise direction, but very fine in the direction of boundary layer growth, in order to capture the rapid physical
changes occurring there. We need very fine cells as we cross the interface, but we can afford the elongation in the streamwise direction in order to model a significant downstream length. The simulation results are shown in Figure 5.11 along with the predicted analytic surface profiles of Tillet for these cases.

The results show excellent agreement with theory for these larger downstream distances. This implies that the method of increasing the domain length by keep the relative grid resolution constant is appropriate and successful, in spite of the presence of very large cell aspect ratios. This technique is adopted and employed for later simulations. Based on the above tests, we are confident in the ability of the CMGFD code to incorporate gravitational effects into planar jet simulations for various downstream distances.

5.3 2-D Jet with Surface Tension

We compute a jet profile with surface tension forces present, while neglecting gravitational effects. The mesh of Section 5.1 is used for the simulation with water and air as the working fluids, and a Reynolds number of $Re = 200$. When investigating surface tension effects, it is useful to introduce the Weber number, $We$, which represents the ratio of inertial forces to surface tension forces and is defined as

$$We = \frac{\rho_l U_m^2 h}{\sigma}$$

(5.11)
Here, $\rho_l$ is the density of the liquid phase, $U_m$ is the mean velocity at which the liquid exits the channel, $h$ is the width of the channel, and $\sigma$ is the surface tension coefficient. For a water-air interface, the surface tension coefficient is $\sigma = 0.073 \text{N/m}$. Using these physical properties and the dimensions of the jet from Section 5.1, the Weber number of the present test case is $We = 0.5$, implying that neither inertia or surface tension will clearly dominate the flow, and that surface tension effects are important and should be visible in the present simulation.

Surprisingly, the *CMGFD* code is unable to produce a result for the $Re = 200$ jet. The computed profiles are shown in Figure 5.12. According to the simulation, the force due to surface tension gives rise to a strong deformation of the jet profile, which ultimately leads to convergence problems in the numerical solution. From the results presented here and others determined from numerical experiments, a strong deformation occurs near the upper edge of the jet inlet. Once the deformation occurs, there are regions of high curvature present, which are then further acted upon by surface tension, causing more deformation and perpetuating the problem. In an effort to remedy the issue, several options are investigated. First of all, much finer meshes and smaller time steps are used (at the cost of significantly higher run times) to better understand the nature of the problem. There is no improvement in the solution quality and behavior. Next, the surface tension force is manually removed from the first column of cells adjacent to the jet inlet, in an attempt to try and alleviate any strong curvature regions near the inlet that are causing artificial deformation of the jet. This approach does not yield any successful results, and in fact leads to non-physical vertical spreading of the jet well above and beyond the jet exit plane. This eventually leads to a divergence of the solution. The density ratio of the fluids is also decreased further to try and improve the behavior. Unfortunately, reducing the ratio to values as low as 50 does not give any positive results, and the same behavior is present. Finally, the surface tension forces are both underrelaxed and gradually introduced into the simulation during a later time to try and avoid any transient phenomenon causing the above behavior. This does not lead to stable, physical solutions for the simulation as would be expected in this low Weber number case. There is no physical reason for divergence in the solution and the failure of the method is an unknown numerical artifact. One possible explanation is from consideration of the momentum equations. For the $Re = 200$ case, the Weber number is $We = 0.5$, meaning that the surface tension effects are highly significant. The source term present in the equations may be relatively large compared to the inertial and viscous terms. The line Gauss-Seidel solver used in the *CMGFD* code may not be able to accurately solve the system of equations due to the presence of this large source term.
Figure 5.12: Planar jet with surface tension at various times, $Re=200$
In an effort to further investigate this behavior, the simulation is repeated for a higher Reynolds number jet. In the previous section the effects of gravity are diminished by the increased inertia of a higher Reynolds number, and using a higher Reynolds number jet also causes the Weber number to increase, giving more dominance to the inertial forces in the flow. The case of a $Re = 1100$ jet is simulated here in the presence of only surface tension forces (without gravity). At this Reynolds number, the jet outlet velocity is $U = 1.0 \text{m/s}$, and the corresponding Weber number is $We = 15.0$. The higher Weber number indicates the surface tension forces are less prevalent here than in the $Re = 200$ case. It stands to reason that any troubles related to the surface tension forces in the previous case should be reduced with these modifications.

Before discussing the solution to this case, we mention that several numerical tests on this problem lead to the use of a slightly more refined mesh than is used for the tests in Sections 5.1 and 5.2. For the present case, the previous mesh is too coarse in certain regions, and the solutions generated with surface tension are not of sufficient quality to be used. The mesh is slightly refined in certain areas to make it somewhat more suitable for use. The new grid is shown in Figure 5.13 while its dimensions are listed in Table 5.5. It is worth mentioning that using this refined grid in the $Re = 200$ case does not produce any successful solutions. The results of the test case are given in Figure 5.14

<table>
<thead>
<tr>
<th>side</th>
<th>dimensions</th>
<th>segment</th>
<th>cells</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>west</td>
<td>$x = 0, 0 \leq y \leq 3.96 \times 10^{-4}$</td>
<td>1</td>
<td>9</td>
<td>parabolic $u(y)$</td>
</tr>
<tr>
<td></td>
<td>$x = 0, 3.96 \times 10^{-4} \leq y \leq 5.5 \times 10^{-4}$</td>
<td>1</td>
<td>14</td>
<td>parabolic $u(y)$</td>
</tr>
<tr>
<td></td>
<td>$x = 0, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>13</td>
<td>internal</td>
</tr>
<tr>
<td>east</td>
<td>$x = 0.01, 0 \leq y \leq 5.5 \times 10^{-4}$</td>
<td>1</td>
<td>17</td>
<td>$\frac{\partial u}{\partial n} = 0$</td>
</tr>
<tr>
<td></td>
<td>$x = 0.01, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>16</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>north</td>
<td>$0 \leq x \leq 0.0011, y = 1.65 \times 10^{-3}$</td>
<td>1</td>
<td>34</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>south</td>
<td>$0 \leq x \leq 0.0011, y = 0$</td>
<td>1</td>
<td>34</td>
<td>$\frac{\partial v}{\partial n} = 0$</td>
</tr>
<tr>
<td>west</td>
<td>$x = -0.0011, 5.5 \times 10^{-4} \leq y \leq 1.375 \times 10^{-3}$</td>
<td>2</td>
<td>11</td>
<td>$u = 0$</td>
</tr>
<tr>
<td></td>
<td>$x = -0.0011, 1.375 \times 10^{-3} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>3</td>
<td>$\frac{\partial u}{\partial n} = 0$</td>
</tr>
<tr>
<td>east</td>
<td>$x = -0.0011, 5.5 \times 10^{-4} \leq y \leq 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>13</td>
<td>internal</td>
</tr>
<tr>
<td>north</td>
<td>$-0.0011 \leq x \leq 0, y = 1.65 \times 10^{-3}$</td>
<td>2</td>
<td>11</td>
<td>$u = 0$</td>
</tr>
<tr>
<td>south</td>
<td>$-0.0011 \leq x \leq 0, y = 5.5 \times 10^{-4}$</td>
<td>2</td>
<td>11</td>
<td>$u = 0$</td>
</tr>
</tbody>
</table>

Table 5.5: Dimensions and boundary conditions used in surface tension jet tests

Figure 5.14 shows a comparison of the computed jet profile with the analytical profile of Tillet (no surface tension effects present). Surface tension causes a significant difference in
CHAPTER 5. PLANAR JET SIMULATION

Figure 5.13: Mesh used for jet simulation in surface tension tests

Figure 5.14: Jet with surface tension, compared with Tillet profile, $Re = 1100$
the interface shape than in the previous cases where its effects are excluded. There is a severe contraction of the jet that occurs immediately after the exit plane. After a short distance downstream, the jet settles into an essentially constant profile with little to no contraction as it propagates downstream. The presence of surface tension forces also causes the interface profile to be more smeared than in previous test cases. This is possibly due to the smoothing used for the volume fraction and in the calculation of the curvatures in the subroutines.

There are no analytical results for the case of a two-dimensional planar jet with a parabolic velocity profile and surface tension forces. Duda and Vrentas [8] devise a numerical procedure in which coupled parabolic partial differential equations are iteratively solved in conjunction with a numerical integration procedure to obtain a solution for a jet profile, which includes the effects of surface tension and gravity. The gravitational effects are suppressed easily within their method, giving results for a jet profile with surface tension only. However, the implementation of their procedure is very difficult to couple with the mesh and program used for the current simulation, and it is a complex task to use it to compare results with the those currently obtained by the CMGFD code. Therefore, their method is not used for comparison with the computed results.

Lienhard [26] uses a different approach to obtain a numerical solution for the case of a jet with parabolic velocity profile in the presence of surface tension and gravity. Unfortunately, his results include gravity, which we do not here, and his solution also consists of a set of parabolic PDE's which must be iteratively solved. Once again, this method is very difficult to use for comparison with our results, and is not implemented. His solution was successfully verified against several experimental results, but was found to fail in certain cases. In these tests, the contact angle between the jet and the channel exit lip had a profound influence on the amount of contraction near the exit plane. For the experiments, a jet of liquid issued from a standard thin tube. If the lip of the tube was wetted by the liquid, the effect of the contact angle on the jet contraction was minimal, and the computed profile was in excellent agreement with the experimental result. However, if the lip of the tube was not wetted, there was a high degree of contraction due to the contact angle (similar to that observed in our simulations), and the computed results did not agree with experiment.

It appears there is significant correlation between the jet profile and contact angle for various materials. The current CMGFD program does not include any considerations for contact angle specification where an interface meets a solid wall. We assume that a "natural" contact angle is automatically created by the surface tension subroutines for any specific cases where such a phenomenon might occur. While it appears possible that the contact angle effects cause a different degree of jet contraction [26], it would be a very involved process
to investigate these effects either numerically or experimentally. Although this behavior is most certainly important, the investigation is not performed within this study.

As we cannot implement the numerical solutions of Lienhard [26] and Duda and Vrentas [8], we cannot directly verify the accuracy our results. Based on the simulations from Sections 5.1 and 5.2, we assume the mesh is well enough resolved to accurately capture the behavior of this test case, and we assume that since the high jet contraction in our simulations is experimentally observed, we are computing a physically valid result. This implies that our incorporation of surface tension effects within jet simulations is also valid.

5.4 2-D Jet with Surface Tension and Gravity

The results of the previous sections suggest that the effects of gravity and surface tension are simulated to a sufficient degree of accuracy by our program. We now include both effects simultaneously into the calculation of a planar jet. As mentioned in the previous section, the case of a jet with parabolic velocity profile and surface tension and gravity has an approximate numerical solution derived from the boundary layer equations as calculated by Lienhard [26]. Duda and Vrentas [8] also provide an approximate solution to this problem by use of an ingenious coordinate transformation of the boundary layer equations. While these results could give an estimate on the accuracy of the results produced by the code for jets in the presence of both surface tension and gravity, both methods are very complicated and require the numerical solution of several coupled partial differential equations. Although potentially useful, these equations are too involved to be solved for and compared with our numerical results. Hence, we base the validity of the results on the basis of the previous sections in this chapter. The simulation of jets with surface tension and gravity will proceed through the computation of jet profiles corresponding to experimental cases where surface waves were observed to naturally occur.

5.4.1 Comparison With Experimental Results

One goal of this thesis is to use an appropriate two-phase flow solver to attempt to simulate waves on the surface of a planar jet. The modified CMGFD code has the ability to accurately simulate two-phase flow phenomenon and we attempt to reproduce the experimental results of Söderberg [41] regarding the presence of surface waves on planar jets.

We simulate the cases run by Söderberg for a very thin, essentially two-dimensional water jet in the presence of gravity. The details of the experimental apparatus and set-up
are found in [41]. The experiment consists of a jet aligned vertically in the direction of gravity. The jet has a thickness of 1.1\,mm and a width of approximately 25\,cm, so it is safe to assume it is independent of the spanwise behavior and that the use of a two-dimensional simulation is valid. The jet is run at various mean velocities, ranging from $1.3\,m/s$ to $4.4\,m/s$, with corresponding Reynolds numbers of $1430$ to $4800$, as based on the half-channel height, $a = 0.55\,mm$. A schematic of the set-up is shown in Figure 5.15.

Söderberg experimentally shows that for a water jet with the above thickness and a mean velocity of $1.3\,m/s$ ($Re = 1430$), there are no surface waves present. However, increasing the velocity to $1.5\,m/s$ ($Re = 1650$) and beyond gives rise to naturally occurring surface waves, similar to those shown in Figure 1.4. We simulate the two-dimensional cases corresponding to $U_m = 1.3\,m/s$ or $Re = 1430, \, We = 25.4$ (no waves present), and $U_m = 1.7\,m/s$ or $Re = 1860, \, We = 43.5$ (waves present). The mesh used is the modified version from the surface tension test case in Section 5.3.

In the case of $Re = 1430$ our computed solutions do not show the presence of any waves, as shown in Figure 5.16. The final result is a smooth, steady-state profile, similar to that of the cases in Sections 5.1-5.3. While there is some initial transient behavior as the jet exits the computational domain, the profile does settle into a time-independent shape. As in the previous section, there is a high degree of contraction at the exit plane, due to the presence...
of surface tension effects. The jet again settles into a constant shape which does not further contract on the given length scale. The current simulation was kept at a Courant number of approximately $Co = 0.3$, in order to keep a sharp interface resolution by the CICSAM method.

If the size of the time step is increased, giving a Courant number of $c = 0.75$, traveling disturbances are created on the jet surface, where there are none at a lower time step and Courant number. These disturbances appear to be caused by the air velocity vectors, and they propagate downstream and out of the computational domain. The disturbances are not periodic and occur over several grid cells. These results are shown in Figure 5.17. Due to the fact that the disturbances do not appear at the lower time step, we conclude that they are not a physical occurrence but a numerical artifact present in the simulation. Furthermore, they are not the surface waves that are experimentally observed, as Söderberg reports there are approximately 3-5 wavelengths per centimeter [41], and we observe the same number of wavelengths in a millimeter. These observations indicate that these are false, numerically generated waves, which are not physically realistic.

In the case of $Re = 1860$, Figure 5.18 shows the same sudden contraction that is observed as in the $Re = 1430$ case. There is minimal initial transient behavior and the jet settles to a steady-state constant profile. No surface waves are observed for this case. The time step is chosen so that the Courant number is approximately $c = 0.3$, and no intermediate numerically triggered waves are observed, as was the case for the higher Courant number.
According to experimental results, surface waves should be present for this case, but they are not reproduced through our simulations. This could be due to the relatively short downstream length used for computation in the present simulation. The waves are seen to occur approximately 1-3cm downstream from the nozzle in experiments at this Reynolds number [41]. Therefore, it may be the case that the present simulation cannot resolve the waves because there is insufficient downstream distance to do so.

In order to remedy this problem, we attempt to use the technique discussed in Section 5.2, whereby the physical downstream dimension is increased, but the relative cell sizes are kept constant. Since there is great success for the cases of $Re = 1430$ and $Re = 1860$ in the presence of gravity, we hope the same success is to be had for the present case. However, this grid length extension is not so easily accomplished for the current mesh when surface tension effects are included. There is a significant difference in the transient behavior of the jet as it exits the computational domain. For a grid with a downstream length of 5 diameters
Figure 5.18: Computed profile for experimental jet, $Re = 1860$

(equal to 5.5mm), a large growth appears at the leading edge of the jet, as shown in Figure 5.19.

Figure 5.19: Transient growth for extended domain jet

The further the grid length is increased without adding extra cells, the larger the growth
becomes. This is a problem at the downstream exit of the domain. As discussed in Section 5.1, the boundary conditions are modified to stop air from rushing into the domain and preventing convergence. The exit boundary condition is partially composed of a zero-gradient condition large enough to allow the jet to leave the grid, and a rigid wall used to prevent air from re-entering the domain. The wall size is chosen to cover most of the exit plane in order to prevent as much air as possible from re-entering. For different cases of extended downstream length, the growth becomes large enough to impact against the exit wall, causing the calculation to diverge once this occurs.

Another major issue is the behavior of the zero-gradient condition at the exit and the size of the corresponding opening. When using this condition, there is no control over any of the velocity components at the boundary. In most jet cases, the air enters the grid through a small portion of the zero-gradient opening. Once the air re-enters, there is no way to control it through the boundary conditions, even if an exit mass flow rate is specified. In the cases of the previous sections, this inflow is sufficiently reduced by the small size of the opening, and any inflow is minor enough that it is eventually dissipated as the solution converges. In the presence of surface tension, the inflow does not numerically dissipate and is large enough to cause a subsequent divergence of the calculation. This is a significant hindrance to obtaining successful results for the present case of extended grid length. As mentioned, each test case takes several days to run, and it is not until well within a given simulation that the divergence of the flow becomes obvious. In divergent cases, the calculation is rejected and must be started over, which is a significant cost in run time.

Several downstream lengths are tested at the current mesh density. It is found that for lengths lower than 5 downstream diameters, the leading growth does not impact against the exit wall, but the flow of air into the domain causes divergence of the simulation. For lengths greater than 5 diameters, the leading growth is too large to avoid an impact, and the flow diverges. An obvious remedy to the latter case is to increase the size of the opening at the exit. However, this results in the zero-gradient condition causing more air to rush into the grid and prevent convergence. The conclusion is that the current mesh density is insufficient to extend the downstream length when including surface tension effects. The grid resolution must increase when extending the domain length to obtain a converged solution, unlike the case for gravity where the grid is sufficiently resolved for extension. Unfortunately, it appears the surface tension routines require more cells to be added to the grid, causing a drastic increase in program run time.

An increased grid resolution is used for a downstream length of 5 diameters. The mesh consists of 45 cells in the streamwise direction, while all other cell sizes and dimensions are
kept as in Table 5.5. The case of a $Re=1860$ jet with surface tension and gravity is once again computed and the results are shown in Figure 5.20

![Image]

Figure 5.20: Comparison of $Re = 1860$ jet with surface tension and gravity to Tillet's analytical result, 5 diameter downstream length

Clearly, increasing the number of downstream cells produces a converged result. The calculated jet experiences no convergence problems during the transient process, unlike the previous attempts at downstream extension. The transient growth at the leading edge of the jet is significantly reduced by the increase in mesh resolution, and in turn, there is no impact of the jet with the exit retaining wall. As for the profile itself, the same sudden contraction is once again present as in the shorter downstream case. After the initial contraction, the jet again assumes an essentially constant profile. At this extended downstream length, it is evident that the jet does slowly contract as it moves downstream, which is to be expected on physical grounds. The computed jet profile of Figure 5.20 is smooth and at no point during the transient calculation does any indication of the presence of surface waves appear.

For purposes of comparison, the computed profile is shown against the analytical solution of Tillet, to illustrate the qualitative differences in the two solutions. As in the previous test cases, there is a large discrepancy in the profiles near the exit, due to the sudden jet contraction. However, as the jet moves downstream, the analytical and computed results start to approach each other. While we do not expect the profiles to match near the exit plane, experimental results by Middleman and Gavis [27] suggest they should approach the same final-to-initial diameter contraction ratio, irrespective of the fluids used and the
specific value of the Reynolds number, as long as it is above 200. Their experiments suggest surface tension effects only have an influence on the jet profile in the near-field. Since our jet Reynolds number is well above 200, the computed result does tend to imply correct behavior as far as downstream contraction is concerned, and does show a difference in the near field, as is expected based on the above. Although the downstream extension is successful at producing a result that exhibits the correct qualitatively behavior, there is still no evidence of any waves on the jet surface.

5.5 Chapter Summary

This chapter demonstrates the ability of the CMGFD code to simulate various jet cases. A two-dimensional planar jet without surface tension, in both the presence and absence of gravitational forces is handled well by the code. The results for a jet with and without gravity are in very good agreement with available analytical solutions. A significant issue is the choice of boundary conditions used. Walls are used to bound the computational domain and prevent the inflow of air from preventing convergence of the simulation. The exit plane is a combination of walls and a zero-gradient on the velocity and volume fraction. Surface tension effects are implemented and have a dramatic effect on the initial contraction of the jet. It is believed contact angles have an effect on the amount of contraction and the inclusion of a specified value for this angle should be further investigated.

Using a geometrical set-up corresponding to an experimental study of two-dimensional jets and wave disturbances, we attempt to simulate the presence of surface waves on a planar jet. For the cases of $Re = 1430$, and $Re = 1860$, there are no surface waves computed for the single diameter downstream length. Increasing the downstream length up to 30 diameters while keeping the same mesh density for jets with gravity and without surface tension gives solutions that are in excellent agreement with analytical results. However, when including surface tension effects, the same success is not obtained for extension of the downstream distance. Transient effects coupled with the behavior of the zero-gradient exit boundary condition prevent the solution from converging at lower mesh densities. When using surface tension forces, the mesh resolution must be increased to obtain results for longer downstream lengths, which comes at the cost of higher run times. This is prohibitive as the cases currently investigated have very large running times. An increased mesh density is used to obtain a converged result for a 5 diameter downstream length. There is no evidence of surface waves on the jet.
Chapter 6

Summary and Conclusions

6.1 Thesis Summary

6.1.1 Motivation for the current study

The headbox is a critical component in the modern paper making process. The presence of surface waves on planar jets issuing from headboxes is a major concern with regards to the quality of paper produced by conventional methods. Surface waves are believed to arise from instabilities in headbox flows, and subsequently initiate break-up of the planar jet. The break-up causes streaks to form within the jet velocity profile, which lowers overall paper quality. If a numerical simulation of the entire paper making process could be performed, there would be some degree of predictive potential with regards to the effects of jet break-up and paper quality. Therefore, the goal of the current study is to significantly contribute to the task of numerical simulation of industrial headbox flows and jets by incorporating multi-fluid simulation capabilities into an existing fluid-flow research code. More specifically, we investigate and implement suitable numerical methods for two-phase flow simulation within the framework of finite-volume discretization techniques. The multi-fluid code will be used in an attempt to simulate experimentally observed waves on water jets issuing from channels. If the surface waves can be computed for this experimental case, the information and knowledge gained from these simulations can be used in the prediction of planar jets and surface waves from actual headboxes. Therefore, computation and prediction of surface waves will provide a significant initial step in the simulation and understanding of full industrial headbox jets.
6.1.2 Basic Method and Mathematical Considerations

In order to simulate multi-fluid flows, a suitable numerical technique must be implemented. The approach used in this study is known as the Volume of Fluid (VOF) method. In this method, a composite fluid is formed by combining two separate incompressible Newtonian fluids. The composite fluid simultaneously represents both fluids in a single continuum. The interface between the two fluids is denoted by a discontinuity in the density and viscosity of the composite fluid. The spatial and temporal orientation of the single-phase fluids within the composite fluid is described by an indicator function known as the volume fraction, which represents the ratio of volume of liquid in a given finite region to the volume of the region itself. The motion of this indicator field is governed by a convective scalar conservation equation. Within the VOF approach, it is shown that the composite fluid retains the incompressibility and Newtonian stress-strain relationship of the individual constituent fluids. A single set of mass and momentum equations is obtained which, in addition to the volume fraction equation, governs the motion of this composite fluid. These equations, in turn, describe the behavior of both individual fluids.

As the interface between the fluids is no longer treated as an explicit boundary, but simply by a gradient in the physical properties of the composite fluid, the enforcement of boundary conditions becomes problematic. Surface tension effects are incorporated into the VOF method by use of the Continuum Surface Force (CSF) method. In the CSF method, the interface is represented by a thin transition region near the discontinuity. In this region, a volumetric force is constructed that mimics the effect of the surface tension force on the interface, thereby making it possible to include the effects of surface tension in the VOF based simulation method. When using this method to account for surface tension effects, it is shown that the VOF method satisfies the kinematic and dynamic boundary conditions of standard two-phase flow problems. The method is mathematically suitable for computationally solving multi-phase flow problems.

6.1.3 Numerical Techniques

The equations of motion have no general analytic solution and must be solved numerically for the current problem. Standard finite-volume techniques are used to discretize and solve the mass and momentum equations governing the motion of the composite fluid. Special care must be applied when dealing with the discretization and numerical solution of the volume fraction conservation equation. The goal of any numerical multi-phase simulation is to keep a sharply resolved interface between the fluids, and to avoid diffusion in the boundary at
all costs. Several different discretization techniques for the solution of the volume fraction equation are presented and discussed, and the merits and drawbacks of each method are explained. In order to take advantage of the inherent structure of the existing research code, the interface-capturing solution class is used, and the volume fraction equation is represented as a standard convection equation. It is discretized by means of the CICSAM method, which is a high resolution discretization scheme that uses special numerical techniques to maintain a very sharp and well defined interface between the fluids. The current implementation modifies the CICSAM method by adding an additional second-order accurate time discretization scheme as an alternative to Crank-Nicolson method used in the original method. This is a desirable feature, as the Crank-Nicolson method is known to produce solutions that are temporally accurate, but prone to oscillation. The time discretization is further augmented by the inclusion of an implicit-explicit routine that prevents large intermediate Courant numbers, present in the early iterative stages of the implicit solution procedure, from destroying the solution accuracy. This modification is extremely useful to the solution procedure as it allows larger time steps to be used during simulation of steady-state flows. In addition, the predictor-corrector procedure used in the original CICSAM method to correct overshoots in solution profiles is replaced with the use of the multi-dimensional version of the Universal Limiter. This is also desirable, as the predictor-corrector procedure used in the CICSAM method is difficult to implement within the existing framework of our research code, can be computationally expensive to use, and is often unnecessary for structured meshes, which are used within the code.

Surface tension effects are incorporated through use of the Continuum Surface Force (CSF) method. The CSF method uses spatial filtering to smooth the volume fraction field in the transition region near the interface. The volume fraction field must be sufficiently smoothed in this region in order to remove sharp discontinuities and obtain accurate estimations of the interface curvature, which is essential for constructing the volumetric force representing the contribution of surface tension. The current implementation of the CSF method incorporates both Laplacian filters and kernel based convolution filters to perform the smoothing, as opposed to the B-splines used in the original method. The kernel based methods are generally considered more accurate than Laplacian filters, but the option to use either method in the simulations is a desirable feature in the research code. The term representing the surface tension contribution is added to the momentum equations as a standard volumetric source term, and the option to underrelax this term is also included in the code. After implementing the CICSAM discretization and the CSF method, the volume fraction equation is coupled with the mass and momentum equations, and the equations are solved
using an implicit, fully-coupled procedure throughout the entire computational domain to obtain solutions describing the orientation and motion of the fluids.

6.1.4 Validation

A series of standard test cases described throughout the literature are used to verify the proper functioning of the modified code. First, the behavior of our version of the CICSAM method is quantitatively verified through the comparison of computed results of several advection tests with previously published results. The current implementation (using the Universal Limiter and the Crank-Nicolson time method) shows very good agreement with other similar free surface implementations, including the original version of CICSAM. While our version is quantitatively outperformed by the original implementation, the results with the current method are very similar to those of the original and other more accurate methods, and our version greatly outperforms other less accurate methods, as expected.

Surface tension effects are verified by means of a standard test case, whereby the pressure drop across the interface of a bubble of high density fluid immersed in a lighter density fluid is computed. The current CSF method is tested using both the Laplacian filter method and the convolution based method. In tests using different Cartesian meshes, it is found that the convolution method produces unacceptably inaccurate results, while the Laplacian filter gives results that are very suitable but slightly under predict (within 10-15% of expected values). Due to these results, the Laplacian filter is adopted for all smoothing procedures in surface tension calculations for the current study. Using the Laplacian filter for computations of pressure drop across a bubble on a circular mesh reveals that the predicted values are in near perfect quantitative agreement with expected results, suggesting that the CSF method is implemented correctly. The method is also used to simulate the behavior of a square bubble relaxing its shape under the action of surface tension forces. The simulations are in good qualitative agreement with reported results, giving further credibility to the validity of the current CSF implementation.

With the CICSAM scheme and CSF methods tested and operating correctly, the program is used to compute the behavior of two fluids in actual flow fields and qualitatively judge the performance of the code. A Rayleigh-Taylor instability problem making use of gravity and surface tension effects is investigated. The simulation does reproduce the expected qualitative behavior associated with this flow, and also demonstrates the importance of using a suitably fine mesh in order to properly resolve fluid interfaces. The VOF method cannot cope with very thin fluid regions where two fluid interfaces reside within a single cell.
CHAPTER 6. SUMMARY AND CONCLUSIONS

In the case of the current Rayleigh-Taylor flow, there is a thin tail of fluid which trails a falling spike of denser fluid. The tail is sufficiently thin to be contained within a single cell. Within these cells, there are two different fluid interfaces. This causes the VOF method to assign a single, incorrect volume fraction value to these cells. These artificially low values are then propagated throughout the flow field and domain. This behavior has the effect that while the volumes of the original fluids are conserved, the masses the fluids are not conserved because of these incorrectly assigned volume fraction values. Hence, a suitably fine grid must be used to simulate flows in which very thin interfacial regions may be present, such as the Rayleigh-Taylor instability, in order to properly conserve mass throughout the calculation and give a correct description of the fluid-fluid interface.

The sloshing of water in a tank filled with air is computed on a non-orthogonal grid with surface tension and gravitational effects. It is found that the high density ratio that exists between the fluids causes the simulation to diverge. This is due to the behavior of the lighter density gas phase becoming overly erratic and leading to convergence problems. In order to obtain a solution, the density of the air phase is increased by a factor of 2. For sloshing of water in air, this is acceptable, as the water density remains unaltered, and from a physical standpoint, the dynamics of the air and water do not vary much with the modified density. However, this change has a profound effect on the numerical behavior of the solution, as evidenced by a successfully computed result. The decreased density ratio is sufficient enough to diminish the behavior of the air and obtain a converged solution. The expected sloshing behavior is present and does qualitatively agree with available results. The test demonstrates the ability of the code to simulate multi-phase flows on non-uniform and non-orthogonal grids while incorporating gravitational and surface tension effects. The test cases used in the study verify the proper functioning of all aspects of the multi-phase code, and give some insight into the practical usage of the program.

6.1.5 Simulation of Planar Jets

With the multi-phase modifications verified, the code is used to simulate the behavior of planar jets produced in channel flow. The jets are investigated by means of a series of two-dimensional simulations. All cases involve a jet issuing from a channel with a fully developed parabolic velocity profile. This profile is implemented as an input boundary condition in order to reduce computing time.

The first case computed is a jet in the absence of surface tension and gravity at various laminar exit Reynolds numbers. An analytic result for the steady-state behavior of such a
jet is available and is used for motivation and verification of this case. The current two-phase flow method can only compute steady flows by marching forward in time until all features of the solution exhibit no appreciable change. Significant differences exist between the assumptions used in the analytic case and the conditions used for the actual computed solution. The analytic case assumes a viscous jet enters into vacuum at a constant pressure. The modified code cannot set the physical properties of either of the individual phases to zero, so the simulation is run using water and air as the individual fluids, based on the assumption that the presence of the air will have minimal effect on the solution. Furthermore, using constant pressure boundary conditions for the domain causes the simulation to diverge. As an alternative, the domain is bounded by rigid walls and appropriately placed openings by which the jet can exit. The walls are placed at a suitable distance from the jet, and are assumed to have minimal influence on the jet behavior. These modifications to the boundary conditions are made in order to obtain any converged results for the given case.

In spite of the differences between the computed and analytical problems, the computed results are in very good quantitative and qualitative agreement with theory. Solutions produced with the Crank-Nicolson version of CICSAM possess non-physical oscillations, and take nearly twice as long to reach steady-state profiles than for cases run using the three-time level method, which produce no oscillations. Therefore, the three-time level method is used exclusively for all jet calculations. The tests are repeated for higher air densities in order to test the validity of the assumption of the air having minimal effect on the solutions. The computed results are in good agreement for density ratios of 830, corresponding to water and air, down to values as low as 50, suggesting that the assumption is valid. The results are time consuming to obtain due to the time marching that must be performed to compute steady-state solutions and the relatively small time steps that are needed to satisfy the Courant number condition of the CICSAM method. Run times for the fully coupled implicit procedure on moderately dense meshes are on the order of several days, even while using fairly fast computer processors.

Gravitational effects are introduced into the jet simulations. The results for jets of various Reynolds numbers are completely consistent with expectations based on analysis of the corresponding Froude numbers for the given cases. The tests show excellent agreement with appropriate analytic results for cases near to the nozzle exit plane and relatively far downstream from the nozzle. There are no significant numerical difficulties introduced by the inclusion of gravity into the simulations, and the solutions are physically plausible and well behaved.

Surface tension effects are incorporated into the jet simulations. The code is unable to
produce results for low Weber number jet flows, where the neither the inertial, viscous, or tensile forces have a complete dominance over the flow. Several techniques are unsuccessfully applied in an attempt to produce a convergent solution for this case. It is believed that the source term used to represent surface tension in the momentum equations is very large for this case, and the numerical solver used in the code may have difficulties in dealing with it. The tests are repeated using a higher Reynolds number and Weber number and a converged solution is obtained. The resulting jet profiles experience a large contraction near the exit plane, due to surface tension forces. The contraction is assumed to be a result of contact angle effects between the jet and the edge of the channel outlet and is physically plausible, based on reported experimental results.

A two-dimensional jet with gravitational and surface tension forces is calculated for various downstream distances for the purposes of reproducing surface waves experimentally observed in low Reynolds number water jets in air. No such waves are observed through simulations of near-field and extended far-field domains. Numerically induced disturbances are present on jet surfaces for higher values of Courant number. These disturbances are not physical and can be eliminated through the reduction of the time step. The mesh resolution for cases with surface tension and gravity must be increased in order to extended the downstream length to avoid large transient growths which cause a divergence of the solution. All computed jet profiles are smooth and show no indication of transient disturbances at any point during the simulation. The present series of tests is unable to reproduce surface waves on planar jets.

6.2 Conclusions

Based on the results of this thesis, several conclusions are drawn relating to the numerical methods used for the simulation of multi-phase flows, and in particular, planar jets.

- The use of an interface-capturing method is appropriate for simulating the motion of two different fluids with simple or complex interface topologies. The methodology is significantly less complex and computationally less expensive to use than an interface-tracking method, which requires deformable grid capabilities to create the interface between the two fluids, and a separate set of solutions to be obtained for each phase. The Volume of Fluid method is an excellent interface-capturing method that provides a straightforward means of incorporating the effects of two fluids into the structure of existing flow-field solvers. The method is mathematically suitable for the solution of two-phase flows.
• High resolution discretization methods provide a straightforward means of discretizing and obtaining solutions to the volume fraction equation. Geometrically based methods, such as piecewise linear methods, give slightly more accurate and resolved interfaces, but these methods are significantly more complex to implement, and do not treat the volume fraction equation as a standard scalar conservation equation for the purposes of discretization. Instead, they require alternate, specialized techniques and procedures to discretize and solve for the volume fraction. In turn, these methods do not take advantage of commonly used discretization procedures and equation solvers that exist in many codes, such as Gauss-Seidel line solvers, to obtain the motion of the indicator field. High resolution methods do make use of existing solution methods, adding to their ease of applicability and use.

• The CICSAM method is a high resolution method that is easy to use and provides a very sharp interface profile. The method has a built in flux-averaging based on the Crank-Nicolson time discretization. It is found that this feature is not essential to the solution procedure, and the Crank-Nicolson scheme can be substituted for by the three-time level method. The methods give essentially the same accuracy of result, but the three-time level method results in shorter run times and has solutions that are less prone to oscillation. Furthermore, the predictor-corrector procedure used in CICSAM to eliminate any remaining overshoots in solution is also unnecessary for obtaining bounded, sharp solutions on structured meshes. An application of a suitable multidimensional limiter will suffice in keeping the solutions bounded in the few occasions where the CICSAM routine cannot.

• An implicit-explicit time discretization method for the volume fraction equation is essential when using an implicit fully-coupled solver. During the initial iterations of a solution, large intermediate Courant numbers are produced. If CICSAM were to be used to compute the interface with these values, the interface integrity would be compromised. The time step would be lowered so that CICSAM could be used for all iterations. In the current version of the method, the CICSAM method is used when the Courant number is suitably low, and for higher values the Superbee method is used to keep interfaces sharp. The implicit-explicit method allows the Superbee scheme to be used for the initial iterations, where Courant numbers are large and would corrupt the CICSAM method. Once the Courant numbers subside, the method switches over to the CICSAM routine. This allows the sharper CICSAM method to be used more often during a given time step and results in more a sharply defined interface. It also
allows higher time steps to be used than with the CICSAM method alone, which is very helpful in reducing run times when computing steady-state solutions.

- Surface tension effects are incorporated through the use of a Laplacian filter for volume fraction smoothing, and a first-order discretization method, as opposed to the second-order kernel based convolution methods. This result is unexpected, but the Laplacian filter based method gives better quality results, while the kernel based method does not agree with published results. However, the first-order method is sufficient for accurately representing the effects of surface tension in simulations.

- In simulating flows with high density differences between the fluids, it may be necessary to increase the value of the lower density in order to obtain a convergent numerical result. Several test cases fail when density ratios corresponding to water and air are used, but increasing the density of the air 2-4 times yields successful results.

- Boundary conditions used for two-dimensional jet simulation are specially formulated to prevent divergence and allow for computation of the solution. They may not be the most natural choice for reproducing the behavior of a free jet in air. The jets are computed with and without the effects of gravity for various Reynolds numbers, grid densities, and downstream lengths of the computational domain. The solutions agree very well with analytical results, suggesting that the boundary conditions are adequately formulated for use in the simulations. The inclusion of gravitational forces poses no additional difficulties in obtaining converged solutions.

- The presence of the air phase has minimal effect on the jet profiles calculated. The solutions are relatively unaffected for a wide range of air density values.

- Inclusion of surface tension forces into the jet simulation requires an increase in the resolution of the mesh, resulting in higher run times for problems with surface tension. In cases where the tensile forces are significant, the code is unable to produce a result, most likely due to the inadequacies of the equation solver in dealing with the large source terms. Decreasing the significance of the surface tension forces by increasing the Reynolds and Weber numbers results in converged solutions.

- Contact angle appears to have a significant effect on the jet near the exit plane. Surface tension causes a more dramatic contraction of the jet profile than in cases without it, and a different specified contact angle may change the amount of this contraction.
• A large time step size for simulations including surface tension and gravity may cause artificial disturbances to appear on the surface of the jet. A reduction in the time step causes these disturbances to disappear, indicating they are non-physical.

• In order to significantly increase the downstream length of the computational domain for problems involving surface tension, grid resolution must be significantly increased to prevent transient growths from causing divergence of the solution.

• Surface waves cannot be obtained through the two-dimensional simulations that have occurred in this study. However, there needs to be further investigation into the capabilities of the modified CMGFD code with respect to two-dimensional jet simulation in order to have a good understanding of the limitations of the program.

• Computational run times associated with two-phase flow problems on modestly dense grids are very high (on the order of days for standard two-dimensional jet problems). A significant increase in grid density poses a serious issue for the solution of problems with an implicit fully-coupled solver when steady state solutions must be generated by marching forward in time.

### 6.3 Recommendations for Future Work

A number of opportunities for further study have arisen as a result of the creation and testing of the present numerical code. These suggestions should be addressed in future work on the numerical simulation of two-phase flows and computation of surface waves on planar jets.

• The use of a second-order accurate time discretization for the momentum equations may lead to improved results for transient two-phase simulations. The three-time level method provides an alternative to first-order implicit time schemes that is relatively straightforward to implement and gives solutions with enhanced time accuracy.

• Courant number limitations present in all interface-capturing methods require time steps to be relatively small when computing solutions to any multi-phase flow problem. Implicit time discretizations result in very large computer run times for both transient and steady-state problems due to the iterative techniques needed to solve the large system of equations. A well documented alternative to the implicit methods used here are projection techniques and explicit time discretizations for the momentum equations. These options may be worth investigating for different codes. If implicit methods must
be used, a fully-coupled solver is not the ideal method by which to solve multi-phase flow problems. A segregated solver may be favourable as the associated systems of equations are smaller, leading to shorter run times. Furthermore, if a SIMPLE type segregated method is used, a Poisson equation for the pressure can be constructed. In this case, a multigrid procedure can be adopted to dramatically increase grid resolution and improve solution accuracy, while drastically decreasing run times.

- Alternate methods of surface tension implementation should be investigated. The inclusion of surface tension causes increased difficulty in obtaining converged solutions, due to the presence of a potentially large source term in the momentum equations. Gauss-Seidel line solvers can experience difficulty in dealing with these terms. Different methods of formulating the source term and implementing it into the momentum equations may lead to improved convergence behavior when using this type of solver. Also, convolution based methods of constructing this source term should be re-investigated as alternative to the standard techniques found in the CSF method, as they should offer higher-accuracy results. In addition, contact angle effects at the boundary between a fluid interface and solid walls should be addressed, as they can have dramatic influence on solutions obtained.

- This study has only dealt with laminar, constant temperature flows. While adding heat transfer effects and turbulence modeling capabilities will not, in principle, alter the multi-phase solution process, insight into their adaptation to multi-phase flows needs to be obtained.

- The boundary conditions used for simulation of planar jets should be investigated further. An alternative to the bounding walls currently used should be determined in order to provide a more “natural” computational domain by which to simulate the free jet in air. If use of the bounding walls cannot be avoided, enlarging the grid in all directions may reduce any effects the walls may have on the jet, and provide conditions that are more likely to appear in the actual free jet case. Another issue is at the exit of the computational domain. The normal velocity at the exit needs to be controlled to prevent air flow from re-entering the domain and corrupting the calculation. The zero-gradient condition currently used for the velocity at the outlet may not be the most suitable choice, as there is no control over the behavior of any of the velocity components, and hence no way to prevent flow from re-entering the grid.

- Two-dimensional jets should be further investigated for increased downstream lengths.
Shorter streamwise distances may be insufficient to observe the growth of any transient disturbances, and using increased downstream lengths may allow the jet to give rise to surface waves. The possibility of initiating disturbances by perturbations of the jet is also worth investigating. In addition, full three-dimensional simulations should also be attempted to verify the results obtained for two-dimensional jets. In either case, the number of cells used becomes an issue, and especially in the three-dimensional case, where the number of cells will dramatically increase due to the presence of a third spatial dimension. Making use of multigrid or some other convergence accelerator will be essential. Full three-dimensional simulations should also occur for grids that do not include an assumption of vertical symmetry in order to investigate the possibility of unaccounted for three-dimensional effects potentially causing surface waves.
Bibliography


