

IOWA STATE UNIVERSITY

Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and
Dissertations

1993

Numerical simulation of two and three-dimensional viscous free surface flows in partially-filled containers using a surface capturing approach

Franklyn Joseph Kelecy
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Aerospace Engineering Commons](#), [Civil Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

Kelecy, Franklyn Joseph, "Numerical simulation of two and three-dimensional viscous free surface flows in partially-filled containers using a surface capturing approach " (1993). *Retrospective Theses and Dissertations*. 10829.
<https://lib.dr.iastate.edu/rtd/10829>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.



94

13990

U·M·I
MICROFILMED 1994

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.



University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600



Order Number 9413990

**Numerical simulation of two and three-dimensional viscous
free surface flows in partially-filled containers using a surface
capturing approach**

Kelecy, Franklyn Joseph, Ph.D.

Iowa State University, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106



**Numerical simulation of two and three-dimensional
viscous free surface flows in partially-filled containers
using a surface capturing approach**

by

Franklyn Joseph Kelecy

**A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY**

**Department: Mechanical Engineering
Major: Mechanical Engineering**

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

**Iowa State University
Ames, Iowa
1993**

Copyright © Franklyn Joseph Kelecy, 1993. All rights reserved.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	xi
NOTATION	xiii
1. INTRODUCTION	1
1.1 Problem Description	1
1.2 Numerical Simulation of Sloshing Flows	2
1.3 Literature Survey	5
1.4 Scope of the Present Research	8
1.5 Research Objectives	10
1.6 Description of Chapters	11
2. MATHEMATICAL MODEL	13
2.1 Introduction	13
2.2 Governing Equations	15
2.3 Interface Modeling and Weak Solutions	16
2.4 Surface Tension	18
2.5 The Artificial Compressibility Method	23
2.6 Boundary Conditions	25
2.7 Initial Conditions	28
2.8 Nondimensional Equations	28

3. THE FINITE VOLUME METHOD	32
3.1 Introduction	32
3.2 The Grid	33
3.2.1 Structured versus unstructured grids	33
3.2.2 Curvilinear coordinates	34
3.2.3 The control volume	34
3.2.4 Grid structure at boundaries	38
3.2.5 Nonuniform grids	40
3.2.6 Grid specification	41
3.3 Metric Relations	43
3.3.1 Basic relations	44
3.3.2 A metric identity	47
3.3.3 The gradient and divergence operators	48
3.4 Implementation of the Finite Volume Method	49
3.4.1 Finite volume discretization of conservation laws	50
3.4.2 Cell face areas	53
3.4.3 Cell volumes	55
3.4.4 Interpolation at cell faces	55
3.4.5 Gradients at cell faces	59
4. DISCRETIZATION OF THE TWO-FLUID EQUATIONS	64
4.1 Integral Equations	64
4.2 Column Vector Form	66
4.3 Spatial and Temporal Discretization	68
4.4 Flux Evaluation	71

4.4.1	Inviscid flux evaluation	72
4.4.2	Viscous flux evaluation	81
4.5	Linearization	83
4.6	Boundary Conditions	88
4.7	Nondimensional Equations	92
5.	SOLUTION ALGORITHM	94
5.1	Implicit Time Marching Algorithm	94
5.2	Solution Methods for Block-Banded Systems	96
5.3	Final Form of the Discretized Equations	98
5.4	The LU Scheme	100
5.5	The LGS Scheme	104
6.	RESULTS	107
6.1	Introduction	107
6.2	The Two-dimensional Broken Dam Problem	109
6.3	The Two-dimensional Rayleigh-Taylor Instability	126
6.4	The Three-dimensional Broken Dam Problem	143
6.5	Axisymmetric Spin-up in a Spherical Container	150
6.6	Computational Effort	156
7.	SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS .	168
7.1	Summary	168
7.2	Conclusions	170
7.3	Recommendations for Further Research	172
7.3.1	Multidimensional upwind schemes	172
7.3.2	Solution-adaptive grids	173

7.3.3	Surface tension	174
7.3.4	Turbulence modeling	174
7.3.5	Two phase flows	175
BIBLIOGRAPHY		178
APPENDIX A. SOME NOTES ON SURFACE TENSION		187
APPENDIX B. INVISCID FLUX JACOBIANS		190
APPENDIX C. EIGENSYSTEM OF THE INVISCID FLUX JA-		
COBIANS		193
APPENDIX D. APPROXIMATE LINEARIZATION OF THE VIS-		
COUS FLUX		200
APPENDIX E. BODY FORCE LINEARIZATION		205
APPENDIX F. BLOCK COEFFICIENTS AND RIGHT HAND		
SIDE VECTOR		207
APPENDIX G. CHANDRASEHKAR'S ANALYSIS OF THE VIS-		
COUS RAYLEIGH-TAYLOR INSTABILITY		211
APPENDIX H. STEADY STATE FREE SURFACE SHAPES FOR		
SPHERICAL CONTAINERS		213

LIST OF TABLES

Table 3.1:	Upwind-biased interpolation: schemes resulting from selected values of ϵ_1 and ϵ_2	60
Table 4.1:	Schemes resulting from selected values of κ	77
Table 4.2:	Limiter functions applicable to the TVD MUSCL scheme . .	80
Table 6.1:	Case summary for the two-dimensional broken dam problem	114
Table 6.2:	Case summary for Rayleigh-Taylor instability: set #1 . . .	129
Table 6.3:	Case summary for Rayleigh-Taylor instability: set #2 . . .	141
Table 6.4:	Case summary for the three-dimensional broken dam problem	150
Table 6.5:	Case summary for the axisymmetric spin-up problem . . .	154
Table 6.6:	Representative CPU times (HP 9000/730) for selected validation cases	157

LIST OF FIGURES

Figure 1.1:	Illustration of methods for handling free surfaces numerically: (a) surface fitting, (b) surface tracking, (c) surface capturing	4
Figure 2.1:	The general moving container problem	14
Figure 2.2:	Control volume containing an interface between two immiscible fluids	19
Figure 2.3:	Illustration of the contact line and contact angle	27
Figure 3.1:	Structured grid coordinate mapping from physical space to computational space	35
Figure 3.2:	A hexahedral control volume in physical space and computational space	36
Figure 3.3:	Grid structure at boundaries: (a) partial cells, (b) whole cells	39
Figure 3.4:	Nonuniform grid structure: (a) grid points placed at centroid of control volume, (b) cell faces placed midway between grid points	42
Figure 3.5:	Interpolation of data at cell face $i + \frac{1}{2}$	56
Figure 4.1:	Control volume containing an interface between two immiscible fluids	65

Figure 4.2:	Illustration of flux evaluation for upwind-biased MUSCL scheme	73
Figure 6.1:	Schematic of computer program design	110
Figure 6.2:	Illustration of the two-dimensional broken dam problem	112
Figure 6.3:	Surge front position versus time for $a = 0.05715\text{ m}$	116
Figure 6.4:	Column height versus time for $a = 0.05715\text{ m}$	116
Figure 6.5:	Surge front position versus time for $a = 0.1143\text{ m}$	117
Figure 6.6:	Column height versus time for $a = 0.1143\text{ m}$	117
Figure 6.7:	Free surface motion for two-dimensional broken dam problem	119
Figure 6.8:	Velocity field for two-dimensional broken dam problem	121
Figure 6.9:	Photographs of broken dam experiment (from Martin and Moyce, 1952)	123
Figure 6.10:	Comparison of density profiles for first and third order interpolants	124
Figure 6.11:	Comparison of density profiles for various limiters	124
Figure 6.12:	Illustration of the Rayleigh-Taylor instability problem	128
Figure 6.13:	Time history of the free surface motion along the left boundary	131
Figure 6.14:	Time history of the free surface motion along the right boundary	131
Figure 6.15:	Free surface motion for the Rayleigh-Taylor instability : $Re = 28.3$	132
Figure 6.16:	Velocity field for the Rayleigh-Taylor instability : $Re = 28.3$	134
Figure 6.17:	Free surface motion for the Rayleigh-Taylor instability : $Re = 283$	136
Figure 6.18:	Velocity field for the Rayleigh-Taylor instability : $Re = 283$	138

Figure 6.19: Evolution of average perturbation amplitude for several modified Reynolds numbers	142
Figure 6.20: Nondimensional perturbation growth rate versus modified Reynolds number: comparison of numerical results with linear theory .	142
Figure 6.21: Illustration of the three-dimensional broken dam problem .	144
Figure 6.22: Surge front position versus time for $a = 0.05715\text{ m}$	145
Figure 6.23: Column height versus time for $a = 0.05715\text{ m}$	145
Figure 6.24: Free surface motion for the 3D broken dam problem	146
Figure 6.25: Symmetry plane velocity fields for the 3D broken dam problem at selected times	148
Figure 6.26: Cross channel velocity fields for the 3D broken dam problem at selected times	149
Figure 6.27: Illustration of axisymmetric spin-up problem	151
Figure 6.28: The bispherical coordinate system	153
Figure 6.29: Cross sectional view of grid used in axisymmetric spin-up calculations	155
Figure 6.30: Free surface motion for the axisymmetric spin-up problem: 30 rpm case	158
Figure 6.31: Velocity field for the axisymmetric spin-up problem: 30 rpm case	160
Figure 6.32: Comparison of computed wall free surface position with the steady-state, analytical solution: 30rpm case	162
Figure 6.33: Free surface motion for the axisymmetric spin-up problem: 60 rpm case	163

x

- Figure 6.34: Velocity field for the axisymmetric spin-up problem: 60 rpm
case 165
- Figure 6.35: Comparison of computed wall free surface position with the
steady-state, analytical solution: 60rpm case 167
- Figure H.1: Illustration of rotating spherical tank 214

ACKNOWLEDGEMENTS

I would first like to thank the Air Force Office of Scientific Research for their generous support of my research under Grant #AFOSR-0403, and the he NASA Ames Research Center at Moffett Field, California, for providing access to the supercomputers at the Numerical Aerodynamic Simulation (NAS) facility. Additional support for my graduate studies was provided by the Department of Mechanical Engineering at Iowa State University.

I owe a great debt of thanks to my advisor, Dr. Richard Pletcher, for his encouragement, guidance, and patience as this work evolved. His suggestions helped shape many of the novel ideas put forth in the present work. Moreover, he helped maintain my funding during the course of my graduate studies, and for that I will always be grateful. I would also like to thank Drs. Ted Okiishi, Bruce Munson, Joseph Prusa, John Tannehill, Gerald Colver, and Richard Hindman for serving on my examining committee. I would be remiss if I did not also acknowledge the many students and professors (too many to mention here) who have contributed in their own way to my academic experience at Iowa State.

Finally, I could not complete these acknowledgements without mentioning the love and encouragement I have received from my parents, Thomas and Mickalina Kelecy, my wife, Dianne, and daughter, Sarah. With the completion of my academic

career, I hope that I can now dedicate myself more fully to being a good husband and father.

NOTATION**Roman Symbols**

a	length scale associated with broken dam problem
a_i	interfacial area concentration
A	perturbation amplitude
$[A_i]$	inviscid Jacobian matrix
B	source term
\tilde{B}	body force acceleration vector
C	curve bounding discontinuity surface Σ
$[D_i]$	dissipation matrix in numerical inviscid flux
$[D_v]$	approximate viscous Jacobian matrix
e	arc length ratio
f	interpolation factor
f_i	function describing a free surface
F	scalar function used in VOF method
\vec{F}_s	surface tension force
\vec{F}	flux hypervector
\mathbf{F}	flux-area product, $\vec{F} \cdot \vec{S}$
Fr	Froude number

\vec{g}_e	gravitational acceleration
$ g_e $	
G	gradient parameter used in upwind-biased interpolation
$[\mathbf{G}]$	source term Jacobian matrix
H	height
$[\mathbf{H}]$	source term matrix
$\hat{i}, \hat{j}, \hat{k}$	Cartesian unit vectors
$[\mathbf{I}]$	identity matrix
$[\mathbf{I}_0], [\mathbf{I}_\theta]$	diagonal matrices used in coupled formulation
J	metric Jacobian
L	length
Δl	arc length
n	perturbation growth rate
\hat{n}	unit surface normal vector
\hat{N}	unit vector normal to tangent vector of curve C
N_{cv}	total number of control volumes
p	thermodynamic pressure
P	p/β
\mathbf{Q}	column vector of primitive variables
r^l	gradient ratios used in limiter functions
\vec{r}	position vector (non-inertial frame)
R	spherical tank radius
\vec{R}	position vector (inertial frame)
\mathbf{R}	residual

Re	Reynolds number
S_c	surface area bounding container volume Ω_c
\vec{S}	surface area vector
t	physical time
$\bar{\bar{T}}$	stress tensor for an incompressible fluid
u, v, w	Cartesian components of the velocity vector
\mathbf{U}	column vector of conserved variables
V	velocity magnitude
\vec{V}	velocity vector relative to non-inertial frame
W	width
\vec{W}	velocity of discontinuity surface σ
We	Weber number
x, y, z	Cartesian coordinates (non-inertial frame)
X, Y, Z	Cartesian coordinates (inertial frame)

Greek Symbols

α	interpolation parameter or coefficient in initial velocity field equation
β	pseudocompressibility parameter
δ, Δ	discrete differences
Δ	Dirac delta function
ϵ_1, ϵ_2	parameters used in upwind-biased interpolation
ξ, η, ζ	computational coordinates
θ	weighting factor used in two-level trapezoidal scheme

θ_c	contact angle
κ	mean surface curvature
λ	eigenvalues of inviscid flux Jacobian matrix
λ_p	perturbation wavelength
μ	dynamic viscosity
ν	kinematic velocity
$\hat{\nu}$	unit vector normal to discontinuity surface σ
ρ	density
σ	surface tension
$\sigma^\epsilon, \sigma^\eta, \sigma^\zeta$	spectral radii
Σ	discontinuity surface
τ	pseudo time
ϕ	generic variable
Φ^l	limiter functions
$\vec{\omega}$	angular velocity vector
Ω	control volume
Ω_c	volume of container

Superscripts

i, j, k	main grid point indices
l	vector component index or control volume face index
L	associated with “left” running waves
m	pseudo-time index

n	physical time index
p	iteration index
R	associated with “right” running waves
*	denotes nondimensional variables

Subscripts

i, j, k	indices associated with ξ, η, ζ coordinates
i	inviscid or interfacial
v	viscous
0	associated with initial conditions
$1, 2$	denote variables associated with fluids 1 and 2

Other Symbols

$[M]$	coefficient matrix
$\mathcal{O}(\dots)$	order operator
∇	del operator
∇_s	surface gradient operator

1. INTRODUCTION

1.1 Problem Description

The motion of fluids within partially-filled containers has been the focus of much study by scientists and engineers. This interest has been due, in large part, to its importance in many practical applications. For example, civil engineers and seismologists have actively studied the effects of earthquake-induced fluid motions on oil tanks and water towers (Cooper, 1960). In recent years, aerospace engineers have been concerned with the effect of fluid sloshing within propellant tanks on the stability of aircraft, rockets, and satellites (Abramson, 1966). All of these applications seek container designs which minimize the amplitude of fluid forces for a given range operating conditions. Typically, this is accomplished by either modifying the natural frequency of the fluid-container system or by introducing baffling devices to damp the fluid motion.

The present study was motivated by research conducted at Iowa State University on fluid sloshing within satellite propellant tanks (Baumgarten, Flugrad, and Prusa, 1989; Baumgarten, Flugrad, and Pletcher, 1990; Baumgarten, Flugrad, and Pletcher, 1991). The goal of this research was to investigate the unstable, nutational motions exhibited by spin-stabilized communications satellites. To study this problem, a test rig was constructed to examine the behavior of two fluid-filled spherical tanks spun

about a vertical axis (Cowles 1987; Anderson, 1988). Experiments conducted with this test rig helped establish the relationship between key system parameters (e.g. tank position, fill ratio, and rotational speed) and the dynamic stability of the test rig.

In addition to the experiments, a mathematical model of the kinematics and dynamics of the two tank system was developed using a two degree-of-freedom pendulum to simulate the sloshing liquid (Obermaier, 1988). While this model yielded satisfactory results for many cases of interest, it was recognized that improved accuracy could only be obtained through a better understanding of the fluid motion. This recognition led to the idea of using a computational fluid dynamics (CFD) simulation of the liquid motion to augment the system model. The research reported in this thesis is largely an outgrowth of this effort.

1.2 Numerical Simulation of Sloshing Flows

Despite the power and capacity available in present day computers, simulating fluid sloshing within a partially-filled, moving container is still a challenging problem. One of the principal reasons for this is that the flowfield is inherently three-dimensional and unsteady. Numerical solutions of such flows involve setting up a system of grid points within the domain of interest, formulating a discrete representation of the equations of fluid dynamics at each point, and solving the resulting system of equations over a given time interval (or time step). Since the time step must be kept small in order to maintain both the accuracy and stability of the solution, a large number of time steps is required to complete a typical transient calculation. As a result, such calculations can consume enormous amounts of computer memory,

CPU time, and disk storage.

Another difficulty which arises in the sloshing problem is the presence of one or more free surfaces. In the present context, a free surface can be defined as the interface between the liquid and a secondary fluid (usually a gas) which occupies the remaining space within the container. Since the position of the free surface is not known *a priori*, it must be determined as part of the flowfield solution. The free surface motion may also be influenced by various interfacial phenomena, such as surface tension and mass transfer.

Numerical approaches for handling flows with free surfaces can be grouped into three broad categories: surface fitting methods, surface tracking methods, and surface capturing methods. The surface fitting method (illustrated in Fig. 1.1(a)) solves for the flowfield within the liquid region only, the free surface being placed at the boundary of the computational domain. The motion of the free surface is accounted for by using a coordinate transformation which maps the moving, body-fitted coordinate system in physical space to a fixed, uniformly-spaced coordinate system in computational space. If the free surface becomes highly distorted, a new mesh may have to be generated in order to prevent grid singularities and highly skewed grid point distributions.

The surface tracking and surface capturing methods (illustrated in Figs. 1.1(b) and 1.1(c)) avoid the grid-related problems associated with surface fitting methods by employing a grid which is fixed relative to the container, and defining the location of the liquid and gas regions relative to the fixed grid. This practice requires an enlargement of the computational domain to encompass the entire container since the free surface can potentially occupy any position within the fixed grid.

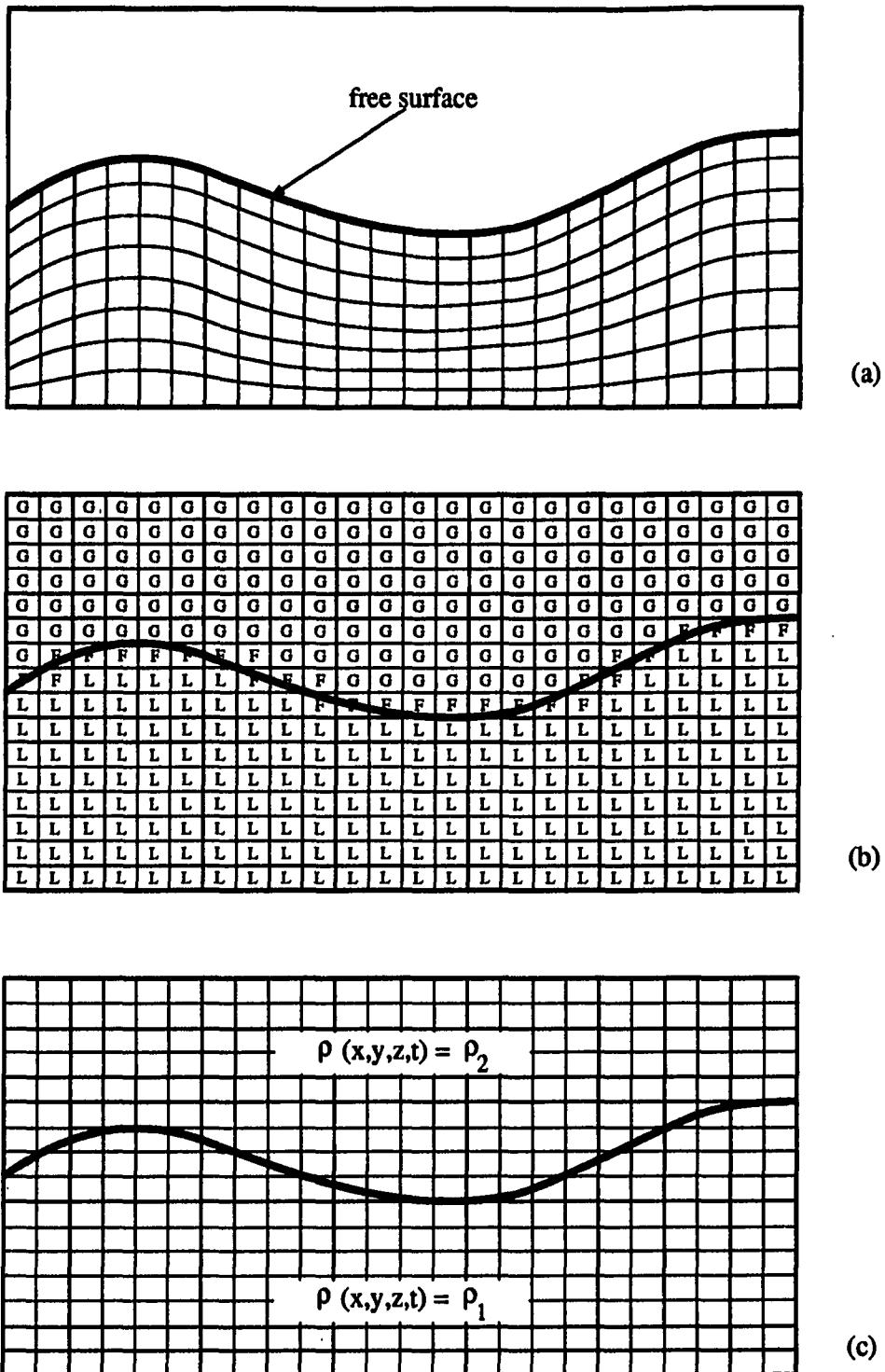


Figure 1.1: Illustration of methods for handling free surfaces numerically: (a) surface fitting, (b) surface tracking, (c) surface capturing

The principal difference between the surface tracking and surface capturing schemes is the manner in which the location of the liquid is identified. As shown in Fig. 1.1(b), surface tracking methods discretize the computational domain into non-overlapping cells which are then “flagged” depending on whether the cell contains all liquid (L), all gas (G), or the free surface (F). The governing equations are then solved in the liquid and free surface cells only. The cells must be reflagged at each time step to reflect the new free surface configuration. Surface capturing methods, on the other hand, solve for the flowfield within both the liquid and the gas cells simultaneously. The location of the free surface is “captured” as a discontinuity in the density field, thus eliminating the need for tracking procedures. This idea is similar in spirit to the “shock capturing” approach, which has been used extensively in simulations of high speed compressible flows (Anderson, Tannehill, and Pletcher, 1984).

1.3 Literature Survey

Prior to the advent of CFD, most analyses of free surface flows within partially-filled containers employed simplified potential flow models and data from experiments (Cooper, 1960). Most often, the investigators were interested in identifying the natural frequencies of the fluid-container system for a given container shape, working fluid, fill ratio, and excitation frequency. Much of this early work was summarized by Abramson (1966).

CFD solutions to viscous, free surface flow problems began appearing in the late 1960s. One of the first significant numerical studies was carried out by Harlow and Welch (1965,1966) using their marker and cell (MAC) method. The MAC method

employed massless marker particles as a means of tracking the motion of the liquid within the computational domain. The positions of these particles were updated by a simple Lagrangian time integration. Once the new particle positions were known, the cells containing particles were flagged to indicate the presence of liquid. The governing equations were then solved using an explicit time integration scheme to obtain the flowfield solution within the liquid (the influence of the gas being considered negligible). Many investigators have since used the MAC method and its variants to compute a variety of two and three-dimensional free surface flows (Hirt and Shannon, 1968; Chan and Street, 1970; Nichols and Hirt, 1973; Wang and Stuhmiller, 1980; Mader, 1988).

A significant improvement to the MAC method was later developed by Hirt and Nichols (1981). Called the volume of fluid (VOF) method, it identified the location of the liquid through the use of a scalar function $F(x, y, z, t)$ representing the fraction of liquid contained in a cell (0 indicating no liquid, and 1 indicating all liquid). This function was updated using a transport equation of the form

$$\frac{\partial F}{\partial t} + \vec{V} \cdot \nabla F = 0 \quad (1.1)$$

where \vec{V} is the fluid velocity vector. As with the MAC method, only the flowfield within the liquid and free surface cells was computed. The VOF method has since been applied to a wide range of two and three-dimensional free surface flows (Sicilian and Hirt, 1984; Veldman and Vogels, 1984; Partom, 1987; Sicilian and Tegart, 1989).

The development of several free surface capturing approaches occurred at about the same time as the MAC and VOF methods. Early work was conducted by Spalding (1974), Ramshaw and Trapp (1976), and Maxwell (1977). Ramshaw and Trapp

(1976) used a non-conservative form of the momentum and energy equations along with a transport equation for density to model the sloshing of water and steam in a rectangular container. Their approach employed a modified donor-acceptor differencing technique to resolve the phase interface (free surface). Both Spalding (1974) and Maxwell (1977) employed the SIMPLE method for incompressible flows (Patankar, 1980) in conjunction with a particle tracking procedure to determine the location of the free surface. Unlike the MAC method, however, their approach obtained flowfield solutions in both the liquid and the gas simultaneously. Some results for several two-dimensional cases were later reported by Maxwell and Spalding (1987). This method was later modified by Jun and Spalding (1988) to incorporate a simple transport equation to update the density field. Special upwinding and property evaluation procedures were used to minimize the diffusion of the density interface. Several two-dimensional computations were presented to demonstrate the effectiveness of their approach. To the author's knowledge, none of the foregoing methods have been applied to three-dimensional problems.

A variety of alternative numerical schemes for free surface flows have also been reported in the open literature. Most notable among these are the free Lagrangian methods, the particle-in-cell (PIC) method, and mixed Eulerian-Lagrangian methods. In many cases, however, these methods are only applicable to specific geometries (usually two-dimensional) or flow conditions. A comprehensive discussion of these and other methods can be found in the review article by Floryan and Rasmussen (1989).

1.4 Scope of the Present Research

Previous numerical studies of sloshing flows at Iowa State University have employed the surface fitting method (Chen, 1990; Kassinos and Prusa, 1990). Surface fitting methods have the distinct advantage that the free surface position is known explicitly. Knowledge of the free surface position permits the surface curvature and other geometric quantities to be computed directly, which, in turn, greatly simplifies the application of interfacial boundary conditions.

Unfortunately, the need to adapt the grid to the changing free surface shape significantly increases the complexity and computational cost of surface fitting methods. In fact, if a structured grid is employed, a complex free surface topology can make the generation a suitable grid very difficult (if not impossible), especially if surface singularities due to wave breaking and other related phenomena are present. Some of these problems can be overcome by using an unstructured grid. However, schemes which employ unstructured grids are generally more difficult to develop, particularly in three dimensions.

Since the flow problems of interest in the present work could potentially involve very complicated free surface motions, it was decided that either a surface tracking or surface capturing method should be adopted. These methods do not require grid alignment with the free surface, and thus are free of the grid distortion problems inherent in the surface fitting method. In addition, problems involving multiple free surfaces, fluid filling or draining, and wave breaking can be accommodated without any special treatment.

While surface tracking methods have been successfully used to compute a variety of free surface flow problems, they were not chosen for use in the present research for

the following reasons. First, since most surface tracking methods (e.g. the MAC and VOF methods) solve the governing equations in the liquid only, the resulting grid is effectively unstructured. For example, in the two dimensional grid shown in Fig. 1.1(b), surface cells may have only one or two neighbors, depending on the orientation of the free surface. This problem can be overcome by using an uncoupled, hybrid explicit-implicit discretization of the governing equations (Hirt and Nichols, 1981). However, this approach imposes severe time step restrictions on fine grids since the maximum stable time step is directly proportional to the grid size.

Another problem with many surface tracking methods is the ad hoc manner in which the free surface is updated. The VOF method, for instance, employs a relatively complex algorithm to ensure that the proper value of the scalar function $F(x, y, z, t)$ is computed at the free surface cells (Hirt and Nichols, 1981). This complexity is necessary in order to prevent cells from overfilling ($F > 1$) or overdraining ($F < 0$), and to maintain the sharpness of the interface. Such algorithms, however, can be difficult to apply in three dimensions (Partom, 1987).

It should also be noted that most forms of the MAC and VOF methods described in the literature employ approximations of Eq. (1.1) which are only first order accurate in time. While first order time accuracy may be sufficient for some problems, second order formulations provide both better accuracy and enhanced robustness for a given grid size.

Finally, most surface tracking and surface fitting methods assume that the secondary fluid has no influence on the motion of the liquid. If the density of secondary fluid is much smaller than the density of the liquid (for example, water and air), then this assumption is valid. However, this need not be the case for other fluid

combinations. Moreover, the motion of the secondary fluid may be of interest even if the density difference is large.

Many of the deficiencies cited above for surface tracking methods can be overcome by surface capturing methods. Specifically, since surface capturing methods solve the governing equations in both fluids, it is easier to devise fully-coupled, implicit discretizations on structured grids. Such discretizations are, in general, more robust and have less severe the time step restrictions as compared to the explicit-implicit discretizations used in the surface tracking approach. Surface capturing methods can also employ consistent formulations of the mass and momentum fluxes within each cell, and thereby avoid the need for ad hoc free surface treatments.

1.5 Research Objectives

The goal of the present research is to investigate the effectiveness of the surface capturing approach as a means of computing complex two and three-dimensional free surface flows within moving containers. Perhaps the most novel feature of this work is the use of the artificial compressibility method (Chorin, 1967) in conjunction with the governing equations for a viscous, incompressible, variable density flow to develop a conservative, fully-coupled numerical algorithm. This framework permits the application of many of the high resolution upwinding schemes which have been developed in recent years for high speed compressible flows. Moreover, recent studies have shown that artificial compressibility-based methods are competitive with other approaches for computing unsteady, incompressible flows (Merkle and Athavale, 1987; Rogers, Kwak, and Kiris, 1989; Pan and Chakravarthy, 1989).

To demonstrate and validate the present surface capturing method, the following

problems will be considered:

- The two-dimensional “broken dam” problem.
- The two-dimensional Rayleigh-Taylor instability problem.
- The three-dimensional “broken dam” problem.
- Axisymmetric spin-up of a partially-filled, spherical container.

1.6 Description of Chapters

The equations which govern the motion of two fluids in a moving container are developed in Chapter 2. Particular attention is paid to the role of surface tension in the formulation. While surface tension is neglected in the problems considered in the present work, there are many applications in which surface tension effects are dominant. Accordingly, the methodology for incorporating surface tension forces into the present formulation is described. The artificial compressibility approach is also discussed at length.

Chapter 3 describes the fundamentals of the finite volume method, which is used to discretize the governing equations defined in Chapter 2. This chapter emphasizes the specific formulas and algorithms adopted in the present work. The application of the finite volume method to the governing equations is detailed in Chapter 4. The equations are first cast into a fully-coupled, implicit form. Methods for constructing the inviscid and viscous fluxes are then examined, followed by a discussion of linearization. The numerical formulation of the boundary conditions is also described.

The solution of the fully-coupled, implicit, linearized system of equations is the subject of Chapter 5. The basic time marching algorithm is defined, followed by

a description of the final forms of the linearized equations and the development of efficient algorithms for solving the block-banded system.

Chapter 6 begins with a brief discussion of the computer programs and computing environment. This information is useful in interpreting the results presented in subsequent sections. The results for the four validation cases described above then are presented. Comparisons are made between the numerical solutions and available analytical results or experimental data.

Chapter 7 summarizes the accomplishments of this work, and suggests possible improvements in the present methodology, including multidimensional upwind schemes, solution adaptive grids, and surface tension. Extensions of the present approach to turbulent and two phase flows are also discussed.

2. MATHEMATICAL MODEL

2.1 Introduction

In the free surface capturing approach, a flowfield solution is sought within both the liquid and gas regions of the computational domain. A mathematical model of this two-fluid system must therefore be able to simulate the physics in both regions in a consistent manner. The development of such a model is the subject of this chapter.

Given the complexity of the general free surface flow problem, it is necessary to introduce some simplifying assumptions in order to make the resulting formulation tractable. Some basic assumptions used in the present work are listed below:

- The two fluids are immiscible.
- Both the density and viscosity are constant within each fluid.
- The flowfield is incompressible, laminar, and isothermal.

The general class of problems under consideration is illustrated in Fig. 2.1. A container of volume Ω_c and surface area S_c is defined with respect to a local Cartesian coordinate system (x, y, z) which is fixed relative to the container. This local coordinate system is assumed to be moving relative to an inertial coordinate system (X, Y, Z) . The motion of the local coordinate system is defined by a position vector

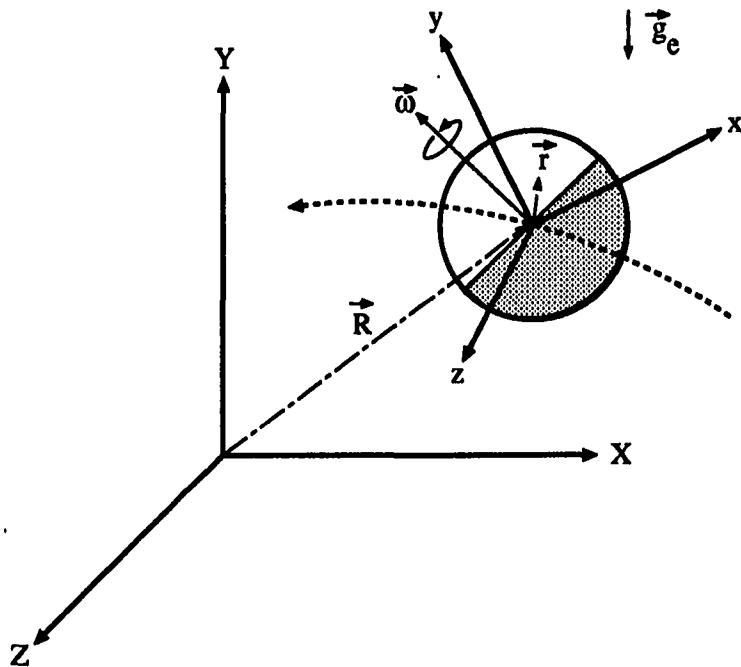


Figure 2.1: The general moving container problem

\vec{R} and an angular velocity vector $\vec{\omega}$, as shown in the figure. Points relative to the local coordinate system are identified by a position vector $\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$, where \hat{i} , \hat{j} , and \hat{k} are the Cartesian unit vectors. In addition to the accelerations introduced by the moving coordinate system, the container may also be influenced by an external gravitational acceleration, \vec{g}_e . It is assumed here that the gravitational acceleration field is constant.

For convenience in the discussions below, the fluid properties in the liquid and gas regions will be denoted respectively by the subscripts 1 and 2. It should be noted that while the primary fluid is usually a liquid and the secondary fluid a gas, the present formulation is applicable to any fluid combination (e.g. liquid-liquid) so long as the foregoing assumptions are satisfied.

2.2 Governing Equations

The mathematical model of the two-fluid system is based upon the system of partial differential equations which govern the motion of a viscous, incompressible, variable density (nonhomogeneous) fluid. These equations can be written in conservation form as follows:

- Conservation of Mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{V} = 0 \quad (2.1)$$

- Conservation of Momentum

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V} - \bar{\mathbf{T}}) = \rho \vec{B} \quad (2.2)$$

- Incompressibility Constraint

$$\nabla \cdot \vec{V} = 0 \quad (2.3)$$

In the above, ρ is the mass density, \vec{V} is the velocity vector relative to the moving coordinate frame

$$\vec{V} = u\hat{i} + v\hat{j} + w\hat{k} \quad (2.4)$$

\vec{B} is the body force acceleration vector (Daily and Harleman, 1966)

$$\vec{B} = \vec{g}_e - \frac{d^2 \vec{R}}{dt^2} - 2\vec{\omega} \times \vec{V} - \frac{d\vec{\omega}}{dt} \times \vec{r} - \vec{\omega} \times (\vec{\omega} \times \vec{r}) \quad (2.5)$$

and $\bar{\bar{T}}$ is the stress tensor for an incompressible fluid

$$\bar{\bar{T}} = \mu \left[\nabla \vec{V} + (\nabla \vec{V})^T \right] - \bar{\bar{I}} p \quad (2.6)$$

where μ is the dynamic viscosity, p is the thermodynamic pressure, and $\bar{\bar{I}}$ denotes the unit dyad.

While Eqs. (2.1) – (2.3) appear unusual at first glance because of the presence of an equation for density, they do in fact constitute a solvable system of equations. Detailed discussions of the mathematical properties of these equations have been presented by Simon (1990) and Antontsev, Kazhilov, and Monakhov (1991). Similar formulations have also been considered by Kataoka (1986) and more recently by Unverdi and Tryggvason (1992). It should be noted that Eqs. (2.1) – (2.3) do not as yet account for any interfacial effects.

To complete the specification of the mathematical problem, Eqs. (2.1) – (2.3) must be supplemented with appropriate boundary and initial conditions. Since the fluid domain is confined to a closed container, all physical boundaries will be impermeable walls. The initial conditions will, of course, depend on the particular problem under consideration. The boundary and initial conditions will be addressed in more detail in later sections.

2.3 Interface Modeling and Weak Solutions

At the interface between two immiscible fluids, experimental observations show that fluid properties are not discontinuous, but instead vary continuously within a narrow transition zone (Edwards, Howard, and Wasan, 1991). Since the width of this transition zone is typically on the order of 10^{-9} m, it is reasonable to adopt the

idealization that the interface is a singular surface (or collection of surfaces) which can be described mathematically by an equation (or set of equations) of the form

$$f_i(x, y, z, t) = 0 \quad (2.7)$$

From an analysis of the kinematics of the interface, it can be shown that the motion of the surface $f_i(x, y, z, t)$ is governed by the equation (Segel, 1977)

$$\frac{\partial f_i}{\partial t} + \vec{V} \cdot \nabla f_i = 0 \quad (2.8)$$

This equation is often referred to as the kinematic boundary condition.

If it is assumed that the fluid interface can be modeled as a singular surface, then, in the absence of viscosity, Eqs. (2.1) and (2.2) reduce to a system of conservation laws which are hyperbolic in both space and time. Using the condition that the free surface is a material interface (and hence moves with the fluid), it can be shown that the inviscid system admits discontinuous solutions (also called weak solutions) wherein both the density and tangential velocity may undergo discrete jumps at the free surface (Jeffrey, 1976). Such discontinuities are referred to as contact discontinuities.

Due to the diffusive nature of viscosity and the molecular mixing which occurs at the interface, solutions to the viscous equations will, in fact, be continuous. However, since the length scales associated with property variations across discontinuities are usually much smaller than the characteristic length of the system under study, one can not, as a practical matter, resolve such variations in a numerical simulation. Consequently, the free surface still appears as a discontinuity in a numerical solution to the viscous equations. This argument has been used to justify the application

of numerical techniques appropriate for the Euler equations (e.g. high resolution upwind methods) in numerical models of the compressible Navier-Stokes equations (Hirsch, 1990).

Following the arguments given above, the present work assumes that the system of equations for nonhomogeneous, incompressible flow can be used to numerically compute free surface flows by “capturing” the free surface as a contact discontinuity in the density field. Specifically, it is asserted that the numerical solution of Eqs. (2.1) - (2.3) for a system containing one or more free surfaces will converge (in the inviscid limit) to the correct weak solution. Since the free surface is “captured” as a discontinuity in the density field, the need for special procedures for tracking and updating the free surface is eliminated.

It should be noted that the concept of obtaining weak solutions to Eqs. (2.1) - (2.3) for the purpose of modeling two-fluid flows is not new. In particular, Kataoka (1986) developed a theoretical framework for this idea in the context of two phase flows. In his work, the differential equations governing the motion of a general two phase system were derived based on the use of generalized functions or distributions. He then showed that if interfacial phenomena were neglected, the equations for two phase flow had the same form as those for single phase flow. However, the derivatives for the two phase model were defined in the sense of a distribution. A detailed discussion of these ideas can be found in Kataoka’s article (1986).

2.4 Surface Tension

The interface between two immiscible fluids exhibits an elastic-like property where, for any line lying in or bounding the interface, a force perpendicular to the

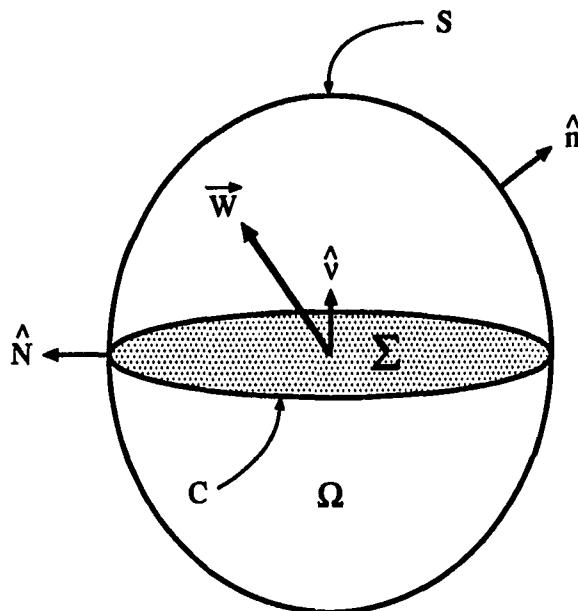


Figure 2.2: Control volume containing an interface between two immiscible fluids

line is exerted in the plane of the interface. This property is called the surface tension, and is the result of attractive forces between fluid molecules. For many problems of interest, surface tension can significantly affect the dynamics of the free surface. In this section, a general form of the momentum jump condition which occurs at a fluid interface due to surface tension will be derived, and a method for incorporating the surface tension into Eq. (2.2) will be discussed.

To begin, consider a material fluid volume Ω which is divided into two subvolumes by a free surface Σ , as shown in Fig. 2.2. The surface area bounding Ω is denoted S , and has an outward-pointing unit normal \hat{n} . The intersection of Σ with Ω is defined by the closed curve C , which also has a unit normal \hat{N} which is orthogonal to both the free surface unit normal $\hat{\nu}$ and a vector tangent to C .

The conservation of momentum for this volume is governed by the following

integral equation (Segel, 1987):

$$\frac{D}{Dt} \int_{\Omega} \rho \vec{V} d\Omega = \int_{\Omega} \rho \vec{B} d\Omega + \oint_S \bar{\bar{\mathbf{T}}} \cdot \hat{n} dS + \oint_C \sigma \hat{N} dC \quad (2.9)$$

For a material control volume divided by a surface of discontinuity, it can be shown that Reynold's transport theorem for an arbitrary scalar or vector quantity ϕ can be expressed as (Aris, 1962)

$$\frac{D}{Dt} \int_{\Omega} \phi d\Omega = \int_{\Omega} \frac{\partial \phi}{\partial t} d\Omega + \oint_S \phi \vec{V} \cdot \hat{n} dS + \int_{\Sigma} \langle \phi \rangle \vec{W} \cdot \hat{\nu} dS \quad (2.10)$$

where $\langle \phi \rangle$ denotes the jump in ϕ across the discontinuity and \vec{W} is the velocity of the interface. Using this relation with Eq. (2.9) gives

$$\int_{\Omega} \frac{\partial \rho \vec{V}}{\partial t} d\Omega + \oint_S \rho \vec{V} \vec{V} \cdot \hat{n} dS + \int_{\Sigma} \langle \rho \vec{V} \rangle \vec{W} \cdot \hat{\nu} dS = \int_{\Omega} \rho \vec{B} d\Omega + \oint_S \bar{\bar{\mathbf{T}}} \cdot \hat{n} dS + \oint_C \sigma \hat{N} dC \quad (2.11)$$

By applying Gauss' theorem individually to each subvolume and adding the results, the surface integrals can be rewritten as

$$\oint_S \rho \vec{V} \vec{V} \cdot \hat{n} dS = \int_{\Omega} \nabla \cdot \rho \vec{V} \vec{V} d\Omega - \int_{\Sigma} \langle \rho \vec{V} \vec{V} \rangle \cdot \hat{\nu} dS \quad (2.12)$$

$$\oint_S \bar{\bar{\mathbf{T}}} \cdot \hat{n} dS = \int_{\Omega} \nabla \cdot \bar{\bar{\mathbf{T}}} d\Omega - \int_{\Sigma} \langle \bar{\bar{\mathbf{T}}} \rangle \cdot \hat{\nu} dS \quad (2.13)$$

Introducing these results into Eq. (2.11) yields

$$\int_{\Omega} \left[\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V} - \bar{\bar{\mathbf{T}}}) - \rho \vec{B} \right] d\Omega + \int_{\Sigma} \langle \rho \vec{V} (\vec{W} - \vec{V}) \rangle \cdot \hat{\nu} dS = \int_{\Sigma} \langle \bar{\bar{\mathbf{T}}} \rangle \cdot \hat{\nu} dS + \oint_C \sigma \hat{N} dC \quad (2.14)$$

Since the free surface moves with the fluid, it follows that

$$(\vec{W} - \vec{V})\Big|_{\Sigma} = 0 \quad (2.15)$$

Thus, Eq. (2.14) reduces to

$$\int_{\Omega} \left[\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V} - \bar{\mathbf{T}}) - \rho \vec{B} \right] d\Omega = \int_{\Sigma} \langle \bar{\mathbf{T}} \rangle \cdot \hat{\nu} dS + \oint_C \sigma \hat{N} dC \quad (2.16)$$

Now, let the subvolumes in Fig. fig:cvolume1 shrink to zero at the interface while Σ remains finite. This implies that the volume integral in Eq. (2.16) vanishes, leaving the relation

$$\int_{\Sigma} \langle \bar{\mathbf{T}} \rangle \cdot \hat{\nu} dS = - \oint_C \sigma \hat{N} dC \quad (2.17)$$

The integral over C in Eq. (2.17) can be transformed into an integral over Σ using the surface divergence theorem (Edwards, Howard, and Wasan, 1991), giving

$$\oint_C \sigma \hat{N} dC = \int_{\Sigma} [\nabla_s \sigma - (\nabla_s \cdot \hat{\nu}) \sigma \hat{\nu}] dS \quad (2.18)$$

where ∇_s is the surface gradient operator. Assuming that σ is constant and noting that the mean surface curvature κ is defined by

$$\kappa = -\frac{1}{2} \nabla_s \cdot \hat{\nu} \quad (2.19)$$

then Eq. (2.18) can be written as

$$\int_{\Sigma} \langle \bar{\mathbf{T}} \rangle \cdot \hat{\nu} dS = \int_{\Sigma} 2\kappa \sigma \hat{\nu} dS \quad (2.20)$$

Since Σ is arbitrary, then at any point on the interface

$$\langle \bar{\bar{T}} \rangle = 2\kappa\sigma\hat{\nu} \quad (2.21)$$

Equation (2.21) is the momentum jump condition for a fluid-fluid interface due to surface tension. It shows that surface tension gives rise to a jump in the normal component of the stress tensor.

If the free surface were placed at the boundary of fluid domain (as is done in the surface fitting approach), then Eq. (2.21) could be used as a boundary condition for the momentum equation. In the present case, however, the free surface lies within the region of interest, thus making the direct application of Eq. (2.21) quite awkward. As a first step towards resolving this problem, one can cast the surface tension force as a field quantity so that it can be incorporated as a source term in the momentum equation. This approach was proposed by Kataoka (1986) for two-phase flow. Slightly different versions of the idea were recently used in CFD applications by Unverdi and Tryggvason (1992) and Brackbill, Kothe, and Zemach (1992). The formulation presented below is based on the work of Kataoka (1986).

Consider a continuous function $f_i(x, y, z, t)$ where $f_i > 0$ in the liquid, $f_i < 0$ in the gas, and $f_i = 0$ at the free surface. Following Kataoka (1986), a local surface tension force $\vec{F}_s(x, y, z, t)$ can be defined as

$$\vec{F}_s(x, y, z, t) = 2\sigma\kappa\hat{\nu}a_i \quad (2.22)$$

where a_i is a quantity called the local interfacial area concentration. Kataoka defines a_i as

$$a_i(x, y, z, t) = |\nabla f_i| \Delta(f_i) \quad (2.23)$$

where $\Delta(t)$ is the Dirac delta function. The local interfacial area concentration can be thought of as a function which represents the free surface area per unit volume at a given point in space. Since the Dirac delta function (and hence a_i) is non-zero only at the free surface (that is, where $f_i(x, y, z, t) = 0$), the surface tension force will only affect the flow in the vicinity of the free surface.

Introducing Eq. (2.22) in the momentum equation yields

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot (\rho \vec{V} \vec{V} - \bar{\bar{T}}) = \rho \vec{B} + \vec{F}_s \quad (2.24)$$

While this specification of the surface tension force is consistent with a field formulation, significant problems remain in attempting to evaluate \vec{F}_s as part of a numerical scheme. Specifically, the surface curvature and the local interfacial area concentration require knowledge of the free surface geometry. This information must be deduced from the flowfield by relating $f_i(x, y, z, t)$ to the density distribution. While flow problems involving significant surface tension forces will not be considered in the present work, some notes how one may approximate \vec{F}_s numerically are provided in Appendix A.

2.5 The Artificial Compressibility Method

In the present formulation, it is assumed that the flowfield is incompressible, as reflected by the inclusion of Eq. (2.3). If density and viscosity of the two fluids are identical, then Eqs. (2.1) – (2.3) reduce to the well-known incompressible Navier-Stokes equations. It is reasonable, therefore, to examine numerical approaches which

have been used successfully for the incompressible Navier-Stokes equations, and apply them (with suitable modifications) to the two-fluid model.

The approach adopted in the present work for solving the two-fluid incompressible system is based on the artificial compressibility method of Chorin (1967). The artificial compressibility method augments Eq. (2.3) with a fictitious time derivative of pressure as follows:

$$\frac{1}{\beta} \frac{\partial p}{\partial \tau} + \nabla \cdot \vec{V} = 0 \quad (2.25)$$

Here, β is a constant (with units of pressure) and τ is a parameter known as the pseudo-time. For convenience, Eq. (2.25) can also be written as

$$\frac{\partial P}{\partial \tau} + \nabla \cdot \vec{V} = 0 \quad (2.26)$$

where a new pressure variable P has been defined as

$$P \equiv \frac{p}{\beta} \quad (2.27)$$

It should be noted that the pseudo time τ has no physical meaning, and, in fact, can be considered an iteration parameter for which

$$\lim_{\tau \rightarrow \infty} \frac{\partial \phi}{\partial \tau} = 0 \quad (2.28)$$

$$\lim_{\tau \rightarrow \infty} \phi(\tau) = \phi(t + \Delta t) \quad (2.29)$$

where ϕ is any flowfield variable. Equations (2.28) and (2.29) imply that the solution at any given future time level $t + \Delta t$ corresponds to a steady state solution in pseudo

time.

Originally, the artificial compressibility method was thought to be applicable only to steady flow problems. For such problems, the entire time dependence was fictitious, with the correct steady state solution being approached asymptotically in pseudo-time as suggested above. More recently, however, several investigators have introduced time-accurate algorithms in which both the physical time t and the pseudo-time τ are marched (Merkle and Athavale, 1987; Rogers, Kwak, and Kiris, 1989). In this case, the solution at a given physical time level is approached asymptotically as a steady state solution in pseudo-time. This type of algorithm, which is often called the dual time-stepping procedure, is employed in the present work as a means of computing time-accurate solutions using the artificial compressibility approach.

2.6 Boundary Conditions

All the flow problems considered in the present work involve the motion of fluids within impermeable containers. Accordingly, only one type of boundary condition arises in the mathematical formulation, namely the viscous, impermeable wall boundary condition.

Experimental evidence suggests that for viscous, Newtonian fluids, the relative velocity at an impermeable wall is zero. This result is known as the no slip condition. For fluid motions relative to the non-inertial coordinate frame (x, y, z) , this implies that at the walls of the container (surface S_c)

$$u \Big|_{S_c} = 0 \quad (2.30)$$

$$v \Big|_{S_c} = 0 \quad (2.31)$$

$$w \Big|_{S_c} = 0 \quad (2.32)$$

A fourth boundary condition for a viscous wall can be derived by evaluating the normal component of the differential form of the vector momentum equation at the wall and applying the no slip condition. The resulting equation can be written

$$\frac{\partial p}{\partial n} = (\nabla \cdot \bar{\mathbf{T}}) \cdot \hat{n} + \rho B_n \quad (2.33)$$

where \hat{n} is a boundary unit normal vector and $B_n = \vec{B} \cdot \hat{n}$. This equation states that the normal pressure gradient at the boundary is influenced by both the viscous stresses and the body force. For orthogonal grids which are sufficiently refined near the wall, it is common to neglect the viscous term in Eq. (2.33), so that

$$\frac{\partial p}{\partial n} \approx \rho B_n \quad (2.34)$$

This approximate form will be employed in the present work.

The fifth and final boundary condition will involve the density. For a single phase incompressible flows, the density is constant everywhere, and, thus, no boundary condition is needed. In the present case, however, the density field is a function of both space and time due to the evolution of the fluid interface. In addition, the intersection of the fluid interface with the walls of the container gives rise to moving contact lines.

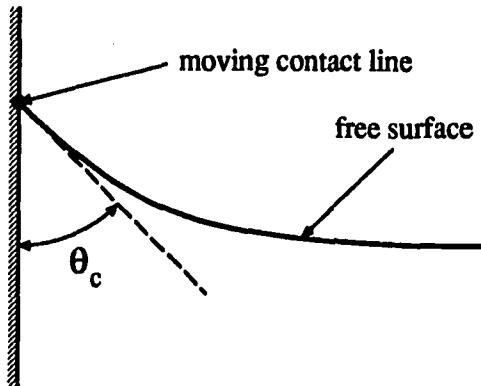


Figure 2.3: Illustration of the contact line and contact angle

The physics of viscous fluid motion in the vicinity of a moving contact line is known to be very complex (Dussan V., 1984). It is found from theoretical and experimental analysis that the shape of the fluid interface is controlled primarily by the contact angle θ_c , as shown in Fig. 2.3. The value of the contact angle depends upon the properties of the two fluids as well as the material properties and finish of the wall. Moreover, the contact angle is, in general, different depending on the direction of motion of the contact line.

In order to make the physics of the moving contact line compatible with the no-slip condition, it is necessary to relax the no-slip condition in the vicinity of the contact line. For modeling purposes, one can identify two regions in the vicinity of the contact line: an inner region where the fluid is allowed to slip at the wall, and an outer region for which the no-slip condition is still applicable. Unfortunately, even with the slip condition at the contact line, the physics are still too complicated to be used in a practical numerical model of wall-bounded free surface flows.

If the length scale associated with the inner region is much smaller than the characteristic length of the system, then it is reasonable to assume that the contact

angle will not significantly affect the flowfield away from the contact line. This assumption will be adopted in the present work. For simplicity, the density values at the wall are obtained by a suitable extrapolation from the interior. It should be noted that this approach is valid away from the contact line, where the local fluid behavior should be consistent with single phase, viscous, incompressible flow.

2.7 Initial Conditions

The initial conditions for the governing equations are given by

$$\rho(x, y, z, 0) = \rho_0 \quad (2.35)$$

$$u(x, y, z, 0) = u_0 \quad (2.36)$$

$$v(x, y, z, 0) = v_0 \quad (2.37)$$

$$w(x, y, z, 0) = w_0 \quad (2.38)$$

where ρ_0 , u_0 , v_0 , and w_0 are prescribed functions defined for all points within Ω_c . As mentioned previously, the specific initial conditions will be problem dependent. Note, however, that the initial velocity field is required to be divergence free by Eq. (2.3). Also, it is important observe that the prescription of initial density and velocity fields implicitly defines the initial pressure field.

2.8 Nondimensional Equations

The governing equations are cast into a nondimensional form in order to introduce important dimensionless groups into the formulation and to permit the flow

variables to be properly scaled (Anderson, Tannehill, and Pletcher, 1984). The standard approach for nondimensionalization consists of selecting reference values for all variables, nondimensionalizing the flow variables using these reference values, and finally incorporating the dimensionless variables into the governing equations. The dimensionless variables used in the present work are summarized below (the superscript * denotes dimensionless variables and the subscript r denotes reference quantities):

$$\begin{aligned} x^* &= x/L_r & y^* &= y/L_r & z^* &= z/L_r \\ u^* &= u/V_r & v^* &= v/V_r & w^* &= w/V_r \\ \rho^* &= \rho/\rho_r & p^* &= p/(\rho_r V_r^2) & \beta^* &= \beta/(\rho_r V_r^2) \\ \mu^* &= \mu/\mu_r & \omega^* &= \omega/(V_r/L_r) & t^* &= t/(L_r/V_r) \end{aligned} \quad (2.39)$$

From the above, it can be seen that four reference quantities must be prescribed for a given problem:

- a reference density ρ_r
- a reference velocity V_r
- a reference length scale L_r ,
- a reference dynamic viscosity μ_r ,

The convention adopted in the present work is to use the properties of the liquid for ρ_r and μ_r . V_r and L_r are selected appropriately for a given problem.

The dimensionless variables are now introduced into Eqs. (2.1), (2.2), and (2.3). After some simplification, the resulting forms of the governing equations can be written

$$\frac{\partial \rho^*}{\partial t^*} + \nabla \cdot \rho^* \vec{V}^* = 0 \quad (2.40)$$

$$\frac{\partial \rho^* \vec{V}^*}{\partial t^*} + \nabla \cdot (\rho^* \vec{V}^* \vec{V}^* - \bar{\bar{\mathbf{T}}}^*) = \rho^* \vec{B}^* + \vec{F}_s^* \quad (2.41)$$

$$\nabla \cdot \vec{V}^* = 0 \quad (2.42)$$

where

$$\vec{V}^* = u^* \hat{i} + v^* \hat{j} + w^* \hat{k} \quad (2.43)$$

$$\vec{B}^* = \frac{\vec{g}_e}{g_e Fr} - \frac{d^2 \vec{R}^*}{dt^{*2}} - 2\vec{\omega}^* \times \vec{V}^* - \frac{d\vec{\omega}^*}{dt^*} \times \vec{r}^* - \vec{\omega}^* \times (\vec{\omega}^* \times \vec{r}^*) \quad (2.44)$$

$$\bar{\bar{\mathbf{T}}}^* = \frac{\mu^*}{Re} [\nabla \vec{V}^* + (\nabla \vec{V}^*)^T] - \bar{\bar{\mathbf{I}}} p^* \quad (2.45)$$

$$\vec{F}_s^* = \frac{2\kappa \hat{\nu} a_i^*}{We} \quad (2.46)$$

Note that ∇ , \vec{r}^* , \vec{R}^* , and a_i^* are all defined in terms of x^* , y^* , and z^* .

Three important dimensionless groups which emerge from the nondimensionalization are the Reynolds number (Re), Froude number (Fr), and Weber number (We). These groups are defined as follows:

$$Re = \frac{\rho_r V_r L_r}{\mu_r} \quad (2.47)$$

$$Fr = \frac{V_r^2}{g_e L_r} \quad (2.48)$$

$$We = \frac{\rho V_r^2 L_r}{\sigma} \quad (2.49)$$

The Reynolds, Froude, and Weber numbers indicate, respectively, the relative importance of viscous, gravitational, and surface tension forces for a given flow problem. Note that the influence of these forces diminishes as the values of the corresponding dimensionless groups increase.

For the sloshing problems of interest in the present work, an order of magnitude analysis of the Weber number using typical property values and length scales reveals

$$We \sim \frac{10^3 \cdot (10^0)^2 \cdot 10^{-1}}{10^{-1}} = \mathcal{O}(10^3) \quad (2.50)$$

where the reference velocity has been scaled with $\sqrt{g_e L_r}$. This relatively large value of the Weber number suggests that the surface tension effects will not be significant. Accordingly, all cases considered in the present work will assume $We = \infty$.

Finally, the pseudo-compressibility equation can be nondimensionalized as

$$\frac{\partial P}{\partial \tau^*} + \nabla \cdot \vec{V}^* = 0 \quad (2.51)$$

Note that the pseudo-time is nondimensionalized using L_r/V_r as a reference time scale.

3. THE FINITE VOLUME METHOD

3.1 Introduction

In selecting a numerical approach for solving the equations presented in Chapter 2, there are many factors to consider. Among these are solution accuracy, computational effort, algorithmic complexity, and computer storage requirements. Computational effort is particularly important in time-dependent calculations since the number of time steps required for a given problem can be quite large. One approach which represents a good compromise between the competing goals of good solution accuracy and reasonable computational cost is the finite volume method.

The finite volume method discretizes the computational domain as a system of non-overlapping control volumes. The relevant partial differential equations are integrated over each control volume, yielding a system of integral equations. Applying specific spatial and temporal discretization techniques transforms these integral equations into a set of algebraic equations, where the unknowns represent spatial averages of the dependent variables for a given control volume.

An important feature of the finite volume method is its ability to maintain both local and global conservation of basic quantities such as mass and momentum. While conservation is a desirable property in and of itself, it turns out to be crucial in discontinuity capturing schemes. Specifically, Lax (1954) has shown that a conservative

formulation is required in order for a consistent numerical scheme to correctly predict the speeds and strengths of discontinuities.

This chapter provides a detailed discussion of the finite volume method, with an emphasis on the specific formulas and algorithms used in the present work. Application of this methodology to the discretization of Eqs. (2.1) – (2.3) will be considered in the next chapter.

3.2 The Grid

The grid consists of an organized collection of points to which discrete values of the dependent variables associated with a given numerical model are assigned. These points are located within the computational domain as well as on its boundary. The manner in which the geometric quantities required by the finite volume method are computed is directly related to the underlying structure of the grid points. This and other related issues are discussed in the subsections below.

3.2.1 Structured versus unstructured grids

The arrangement of grid points within the computational domain can be classified as either structured or unstructured. A structured grid places grid points at the intersections of families of curvilinear coordinate lines. This enables a general coordinate mapping to be set up between a base coordinate system (such as the Cartesian coordinate system) and a general curvilinear coordinate system. An unstructured grid, on the other hand, associates each grid point with its immediate neighbor points in an arbitrary manner. While unstructured grids provide greater flexibility in the placement of grid points, numerical formulations using such grids are

generally more complex to develop, especially in three dimensions. For this reason, the present work employs a structured grid.

3.2.2 Curvilinear coordinates

Given a structured arrangement of grid points in three-dimensional space, a curvilinear coordinate system can be defined such that grid points lie at the intersections of three families of coordinate lines. Here, the family of coordinate lines is denoted (ξ, η, ζ) . Choosing Cartesian coordinates (x, y, z) as the base coordinate system, a mapping can be constructed between the Cartesian and curvilinear coordinates such that the computational domain in physical space (x, y, z) transforms to a box in computational space (ξ, η, ζ) . This idea is illustrated in Fig. 3.1. In general, such a mapping will be nonorthogonal.

The regular structure of the grid allows each grid point to be labeled by a unique set of indices (i, j, k) . As shown in Fig. 3.1, these indices are defined from $(1, 1, 1)$ to $(imax, jmax, kmax)$, and are related to the computational coordinates such that i corresponds to ξ , j corresponds to η , and k corresponds to ζ . The total number of control volumes N_{cv} is given by

$$N_{cv} = (imax - 2)(jmax - 2)(kmax - 2) \quad (3.1)$$

3.2.3 The control volume

For a structured, three-dimensional grid, each control volume is usually defined as a six-sided cell. A typical cell in physical space is depicted in Fig. 3.2(a), and its image in computational space in Fig. 3.2(b). The volume and bounding surface

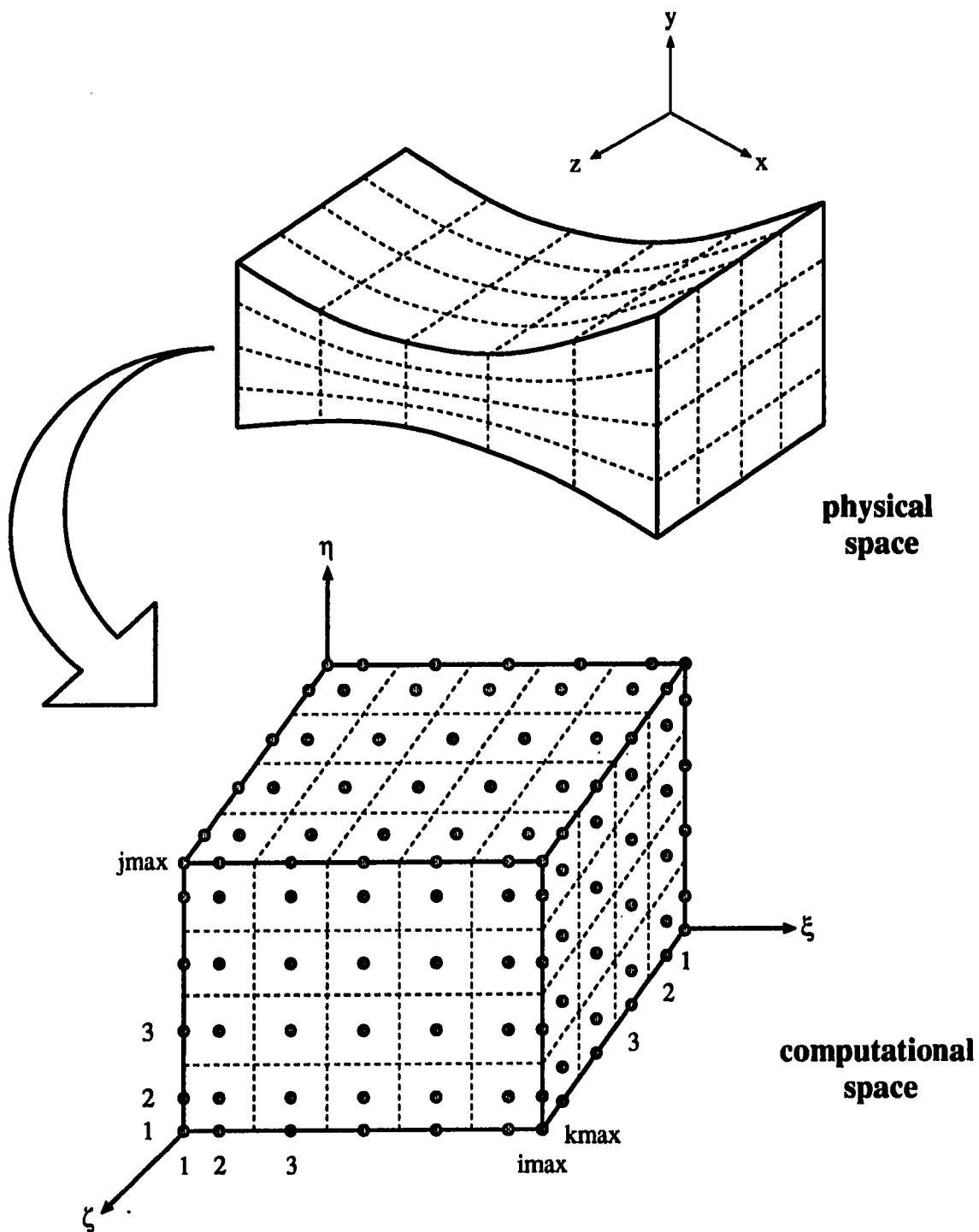


Figure 3.1: Structured grid coordinate mapping from physical space to computational space

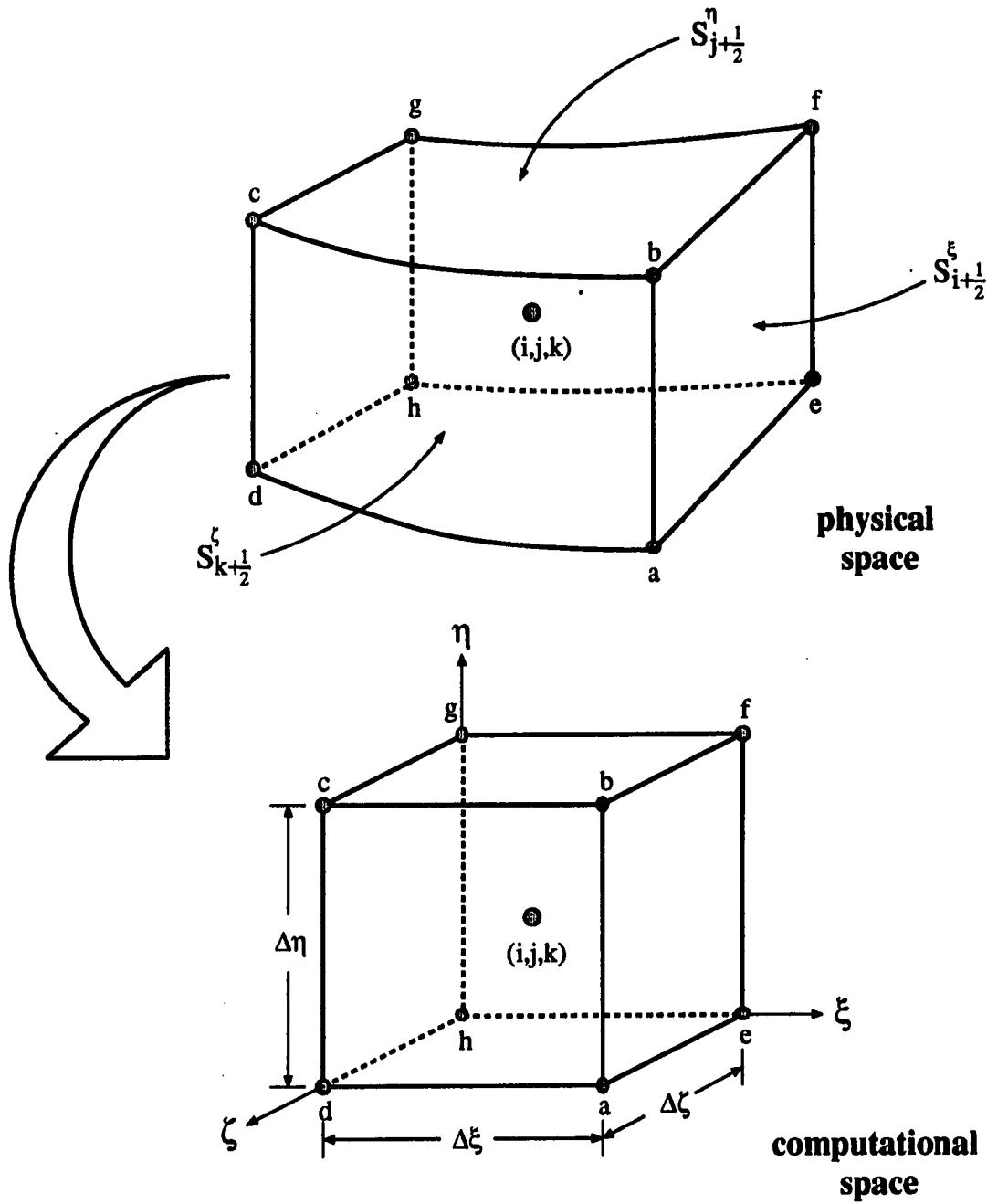


Figure 3.2: A hexahedral control volume in physical space and computational space

area of the control volume in physical space are denoted by Ω and S , respectively, with the six faces of the control volume being identified by the index notation $i \pm \frac{1}{2}$, $j \pm \frac{1}{2}$, and $k \pm \frac{1}{2}$. The present work adopts the idea that each cell encloses the grid points (hereafter referred to as the main grid points), thus allowing a unique set of grid indices (i, j, k) to be associated with each cell.

For a given curvilinear coordinate system, three sets of coordinate surfaces are formed by taking one of the computational coordinates (ξ , η , or ζ) to be constant. It is convenient to associate these coordinate surfaces with the control volume faces by assuming that the $i \pm \frac{1}{2}$ faces are surfaces of constant ξ , the $j \pm \frac{1}{2}$ faces are surfaces of constant η , and the $k \pm \frac{1}{2}$ faces are surfaces of constant ζ . To emphasize this association, the superscripts ξ , η , and ζ will be used in specifying the face areas (e.g. the $i + \frac{1}{2}$ face area is denoted $S_{i+\frac{1}{2}}^\xi$). In addition, the requirement of non-overlapping control volumes implies that adjacent control volumes share a single common face. This property ensures that flux conservation over individual control volumes leads to flux conservation over the entire computational domain.

It can be seen in Fig. 3.2 that the geometry of a cell is characterized by its vertices. For reference, these points have been labeled by the lower case letters a through h . Practical computation of the arc lengths, face areas, and volumes will require knowing the location of these vertices in physical space. In the present work, these data are obtained by specifying the locations of the grid points in physical space, and computing the vertices by interpolation. Before elaborating further on this strategy, however, some additional aspects of grid structure must first be discussed.

3.2.4 Grid structure at boundaries

Because grid points are located on the boundary of the computational domain, a decision must be made with regard to the treatment of control volumes near boundaries. Two possible arrangements are depicted in Figs. 3.3(a) and 3.3(b). In Fig. 3.3(a), a control volume face is placed between the boundary grid point and its adjacent neighbor point. This results in “partial” control volumes at the boundary. The advantage of this approach is that a uniform spacing is maintained between all grid points in computational space. This simplifies finite difference approximations to derivatives at the boundary, as well as other aspects of the discretization. However, special forms of the discretized equations must be derived for all “partial” control volumes.

Shrinking the thickness of each boundary control volume in Fig. 3.3(a) to zero results in the configuration shown in Fig. 3.3(b). In this case, “whole” control volumes completely fill the computational domain. This allows the discretized equations to have the same form for all control volumes, and, since control volume faces lie on the boundary of the computational domain, permits easy application of flux boundary conditions. The obvious disadvantage of this approach is that the spacing of grid points in computational space is not uniform near the boundary.

While either of the above approaches can be used successfully, the second method is more appealing in that the computational domain can be uniformly filled with “whole” control volumes. This approach is adopted in the present work.

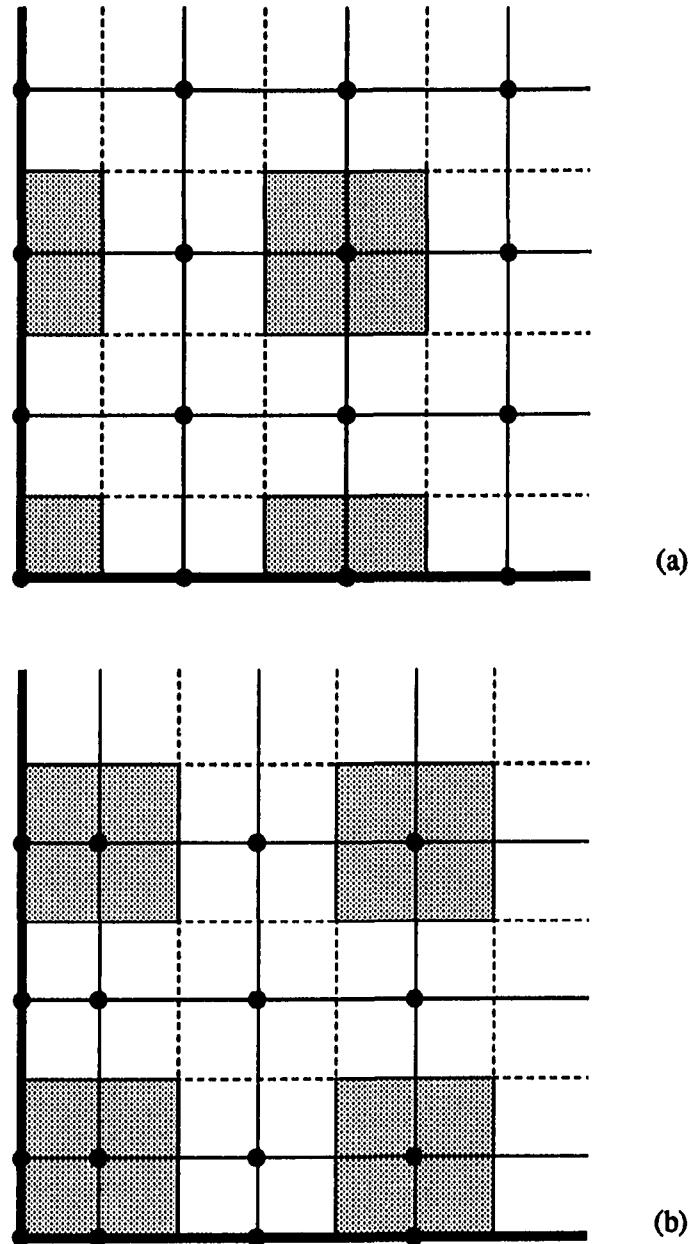


Figure 3.3: Grid structure at boundaries: (a) partial cells, (b) whole cells

3.2.5 Nonuniform grids

If the distribution of the interior grid points is nonuniform in physical space, then a second decision must be made with regard to the position of the control volume faces relative to adjacent grid points. Two possible arrangements are illustrated in Figs. 3.4(a) and 3.4(b).

In Fig. 3.4(a), the grid points are placed at the centroids of the control volumes. This approach has several advantages. First, since volume integrals in the finite volume method are usually approximated as the product of the cell volume and a suitable “average” value of the integrand, the accuracy of volume integral approximations is enhanced. Second, virtually any arrangement of control volumes can be employed without difficulty. In fact, Patankar (Patankar, 1980) notes that discretization of the computational domain can be easily accomplished by first prescribing the locations of the coordinate surfaces (i.e. the control volume faces) and then placing the grid points at the geometric centers of the control volumes. The major disadvantage of this approach, however, is that the control volume face can lie anywhere between adjacent grid points. Consequently, suitable interpolation formulas (which are relatively expensive to compute) are needed to obtain data at the control volume faces.

The alternative to the first approach is to place control volume faces midway between adjacent grid points, as shown in Fig. 3.4(b). The advantage of this approach is that data needed at control volume faces can be obtained (to second order accuracy) using a simple arithmetic mean of data at adjacent grid points. This leads to a simpler formulation and a lower operation count in interpolating cell face quantities. However, as the grid points are generally not located at the centroids of the control

volumes, the accuracy of volume integrals can be reduced.

Again, either of the above approaches can be used successfully in a finite volume formulation. The efficiency of the midway placement of the control volume faces was determined to be the overriding consideration, and was therefore employed in the present work.

3.2.6 Grid specification

The grid specification procedure can now be defined based on the decisions made in the previous sections. The basic steps are presented below.

1. The position vectors of the main grid points are computed using an appropriate grid generation scheme.
2. The cell vertices are determined using suitable the arithmetic mean of adjacent grid points. For example, the vertex \vec{r}_b in Fig. 3.2 is computed by

$$\vec{r}_b = \frac{1}{8}(\vec{r}_{i,j,k} + \vec{r}_{i+1,j,k} + \vec{r}_{i,j+1,k} + \vec{r}_{i,j,k+1} + \vec{r}_{i+1,j+1,k} + \vec{r}_{i+1,j,k+1} + \vec{r}_{i,j+1,k+1} + \vec{r}_{i+1,j+1,k+1}) \quad (3.2)$$

Modified forms of Eq. (3.2) are used at boundaries.

3. Geometric quantities (arc lengths, face areas, cell volumes, etc.) are computed from the known cell vertex data.

Notice that one of the byproducts of this procedure is the placement of control volume faces approximately midway between adjacent grid points.

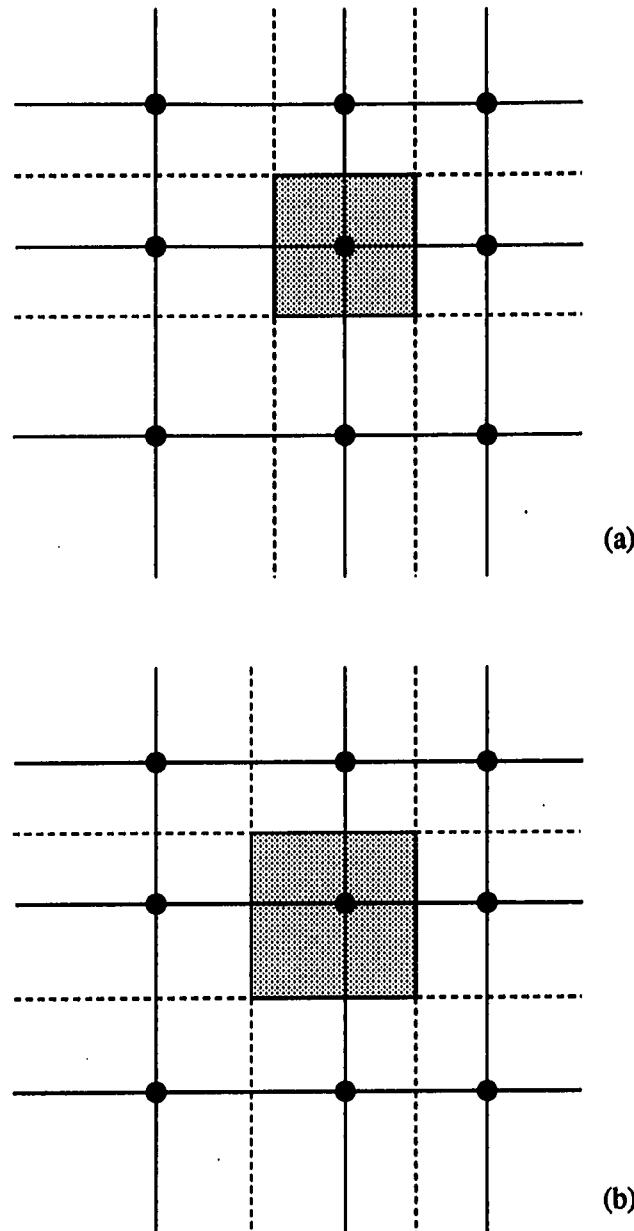


Figure 3.4: Nonuniform grid structure: (a) grid points placed at centroid of control volume, (b) cell faces placed midway between grid points

For computational domains of arbitrary shape, the grid generation task required in Step 1 above is nontrivial. A thorough treatment of the grid generation problem can be found in Thompson, Warsi, and Mastin (1985). In the present work, the geometries considered are simple, and thus do not require elaborate grid generation procedures. For example, a simple Cartesian grid can be generated within a rectangular parallelepiped domain with dimensions $L \times H \times W$ using the following equations:

$$x = x_0 + \xi L / \xi_{max} \quad 0 \leq \xi \leq \xi_{max} \quad (3.3)$$

$$y = y_0 + \eta H / \eta_{max} \quad 0 \leq \eta \leq \eta_{max} \quad (3.4)$$

$$z = z_0 + \zeta L / \zeta_{max} \quad 0 \leq \zeta \leq \zeta_{max} \quad (3.5)$$

Grid generation algorithms employing equations such as those given above were used in the present work to develop specific computer programs for generating the grid point data required by the flow solver. This point will be discussed further in Chapter 6.

3.3 Metric Relations

This section provides the mathematical background needed to develop specific formulas for the geometric quantities employed in the finite volume method. In the interest of brevity, the material is presented without proof (see Thompson, Warsi, and Mastin, 1985 for additional information).

3.3.1 Basic relations

For a stationary grid, a transformation between the Cartesian coordinate system and a general curvilinear coordinate system can be prescribed mathematically as follows:

$$\begin{aligned} x &= x(\xi, \eta, \zeta) \\ y &= y(\xi, \eta, \zeta) \\ z &= z(\xi, \eta, \zeta) \end{aligned} \tag{3.6}$$

This transformation has a unique inverse,

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{3.7}$$

provided the metric Jacobian, J , of the transformation is nonzero. The Jacobian can be written as the determinant

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix} \tag{3.8}$$

Relations linking the Cartesian coordinate system with the general curvilinear coordinate system can be derived using principles from vector calculus and differential geometry (Thompson, Warsi, and Mastin, 1985). These relations can be expressed in terms of the position vector \vec{r} as follows:

$$\frac{\partial \vec{r}}{\partial \xi} = J(\nabla \eta \times \nabla \zeta) \quad (3.9)$$

$$\frac{\partial \vec{r}}{\partial \eta} = J(\nabla \zeta \times \nabla \xi) \quad (3.10)$$

$$\frac{\partial \vec{r}}{\partial \zeta} = J(\nabla \xi \times \nabla \eta) \quad (3.11)$$

$$\nabla \xi = \frac{1}{J} \left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right) \quad (3.12)$$

$$\nabla \eta = \frac{1}{J} \left(\frac{\partial \vec{r}}{\partial \zeta} \times \frac{\partial \vec{r}}{\partial \xi} \right) \quad (3.13)$$

$$\nabla \zeta = \frac{1}{J} \left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \eta} \right) \quad (3.14)$$

$$J = \frac{\partial \vec{r}}{\partial \xi} \cdot \left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right) = \frac{\partial \vec{r}}{\partial \eta} \cdot \left(\frac{\partial \vec{r}}{\partial \zeta} \times \frac{\partial \vec{r}}{\partial \xi} \right) = \frac{\partial \vec{r}}{\partial \zeta} \cdot \left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \eta} \right) \quad (3.15)$$

It should be noted that the vectors $\partial \vec{r}/\partial \xi$, $\partial \vec{r}/\partial \eta$, and $\partial \vec{r}/\partial \zeta$ are tangent to the coordinate lines ξ , η , and ζ , respectively, while the vectors $\nabla \xi$, $\nabla \eta$, and $\nabla \zeta$ are normal to surfaces of constant ξ , η , and ζ , respectively. These metric quantities also

have a physical interpretation as scale factors for differential arc length (dl), surface area ($d\vec{S}$) and volume ($d\Omega$). Specifically,

$$dl^\xi = \left| \frac{\partial r}{\partial \xi} \right| d\xi \quad (3.16)$$

$$dl^\eta = \left| \frac{\partial r}{\partial \eta} \right| d\eta \quad (3.17)$$

$$dl^\zeta = \left| \frac{\partial r}{\partial \zeta} \right| d\zeta \quad (3.18)$$

$$d\vec{S}^\xi = J \nabla \xi d\eta d\zeta \quad (3.19)$$

$$d\vec{S}^\eta = J \nabla \eta d\xi d\zeta \quad (3.20)$$

$$d\vec{S}^\zeta = J \nabla \zeta d\xi d\eta \quad (3.21)$$

$$d\Omega = J d\xi d\eta d\zeta \quad (3.22)$$

Integrating these relations for the arc lengths, cell face areas, and cell volume as shown in shown in Fig. 3.2 gives

$$\Delta l^\xi = \int_{\Delta \xi} \left| \frac{\partial r}{\partial \xi} \right| d\xi \quad (3.23)$$

$$\Delta l^\eta = \int_{\Delta \eta} \left| \frac{\partial r}{\partial \eta} \right| d\eta \quad (3.24)$$

$$\Delta l^\zeta = \int_{\Delta\zeta} \left| \frac{\partial \vec{r}}{\partial \zeta} \right| d\zeta \quad (3.25)$$

$$\vec{S}_{i\pm\frac{1}{2}}^\xi = \int_{\Delta\eta} \int_{\Delta\zeta} \left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right)_{i\pm\frac{1}{2}} d\eta d\zeta \quad (3.26)$$

$$\vec{S}_{j\pm\frac{1}{2}}^\eta = \int_{\Delta\xi} \int_{\Delta\zeta} \left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \zeta} \right)_{j\pm\frac{1}{2}} d\xi d\zeta \quad (3.27)$$

$$\vec{S}_{k\pm\frac{1}{2}}^\zeta = \int_{\Delta\xi} \int_{\Delta\eta} \left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \eta} \right)_{k\pm\frac{1}{2}} d\xi d\eta \quad (3.28)$$

$$\Omega = \int_{\Delta\xi} \int_{\Delta\eta} \int_{\Delta\zeta} J d\xi d\eta d\zeta \quad (3.29)$$

Note that the arc lengths are integrated along coordinate curves which pass through the grid point (i,j,k) , and that the surface area vectors as defined above always point in the direction of increasing ξ , η , or ζ .

3.3.2 A metric identity

The foregoing coordinate transformation satisfies the following metric identity:

$$\frac{1}{J} \left[\frac{\partial}{\partial \xi} (J \nabla \xi) + \frac{\partial}{\partial \eta} (J \nabla \eta) + \frac{\partial}{\partial \zeta} (J \nabla \zeta) \right] = 0 \quad (3.30)$$

Integrating this relation over Ω yields

$$\oint_S \hat{n} dS = 0 \quad (3.31)$$

Physically, Eq. (3.31) follows from the requirement that the surface area S completely enclose the volume Ω . Moreover, Thompson, Warsi, and Mastin (1985) have shown that numerically computed metric quantities must satisfy this identity in order to prevent spurious errors from arising in a CFD solution.

3.3.3 The gradient and divergence operators

In developing discrete approximations to specific terms in the equations of fluid dynamics, one often wishes to express the gradient or divergence of some flow variable in terms of the computational coordinate system. To see how this is done in the present work, let ϕ and \vec{V} denote, respectively, generic scalar and vector functions of space and time. Using relations from computational geometry, it can be shown that the gradient of ϕ , the divergence of \vec{V} , and the curl of \vec{V} can be written as (Thompson, Warsi, and Mastin, 1985)

$$\nabla \phi = \frac{1}{J} \left\{ \frac{\partial}{\partial \xi} [(J \nabla \xi) \phi] + \frac{\partial}{\partial \eta} [(J \nabla \eta) \phi] + \frac{\partial}{\partial \zeta} [(J \nabla \zeta) \phi] \right\} \quad (3.32)$$

$$\nabla \cdot \vec{V} = \frac{1}{J} \left\{ \frac{\partial}{\partial \xi} [(J \nabla \xi) \cdot \vec{V}] + \frac{\partial}{\partial \eta} [(J \nabla \eta) \cdot \vec{V}] + \frac{\partial}{\partial \zeta} [(J \nabla \zeta) \cdot \vec{V}] \right\} \quad (3.33)$$

These relations are often called the conservative forms of the gradient, and divergence. If the metric identity, Eq. (3.30), is applied and the resulting expressions simplified, the following nonconservative forms are obtained:

$$\nabla \phi = \nabla \xi \frac{\partial \phi}{\partial \xi} + \nabla \eta \frac{\partial \phi}{\partial \eta} + \nabla \zeta \frac{\partial \phi}{\partial \zeta} \quad (3.34)$$

$$\nabla \cdot \vec{V} = \nabla \xi \cdot \frac{\partial \vec{V}}{\partial \xi} + \nabla \eta \cdot \frac{\partial \vec{V}}{\partial \eta} + \nabla \zeta \cdot \frac{\partial \vec{V}}{\partial \zeta} \quad (3.35)$$

The metric quantities $\nabla \xi$, $\nabla \eta$, $\nabla \zeta$, and J can be expressed in terms of derivatives of the position vector \vec{r} using the relations given in Section 3.3.1. Hence, discrete forms of either Eqs. (3.32) and (3.33) or Eqs. (3.34) and (3.35) can be developed in a straightforward manner by introducing finite difference approximations of the derivatives of \vec{r} and ϕ with respect to the computational coordinates. Note that Cartesian derivatives can be obtained by simply taking the dot product of the gradient with the appropriate Cartesian unit vectors:

$$\frac{\partial \phi}{\partial x} = \nabla \phi \cdot \hat{i} \quad (3.36)$$

$$\frac{\partial \phi}{\partial y} = \nabla \phi \cdot \hat{j} \quad (3.37)$$

$$\frac{\partial \phi}{\partial z} = \nabla \phi \cdot \hat{k} \quad (3.38)$$

3.4 Implementation of the Finite Volume Method

The metric relations presented in the previous section provide the essential link between the distribution of grid points in physical space and relevant geometric quantities such as arc length, surface area, and volume. The practical use of these relations in a numerical formulation is demonstrated in this section by considering the finite volume discretization of a generic conservation law equation. This discretization will be applied in Chapter 4 to the two-fluid system.

In what follows, it is assumed that the grid has been specified according to the methodology described in Section 3.2.6. The dimensions of each control volume in computational space will arbitrarily be set to unity:

$$\Delta\xi = \Delta\eta = \Delta\zeta = 1 \quad (3.39)$$

This assumption greatly simplifies the discretized equations without sacrificing generality (Thompson, Warsi, and Mastin, 1985). Note that at the boundary points, the width of the cell in the direction normal to the boundary is zero. Accordingly, the corresponding computational space dimension ($\Delta\xi$, $\Delta\eta$, or $\Delta\zeta$) will also be zero.

3.4.1 Finite volume discretization of conservation laws

Consider the generic integral conservation law

$$\frac{d}{dt} \int_{\Omega} U d\Omega + \oint_S \vec{F} \cdot \hat{n} dS = \int_{\Omega} B d\Omega \quad (3.40)$$

where U is a scalar variable, $\vec{F} = \vec{F}(U)$ is a flux function, and B is a source term. Following the general approach developed by Briley and MacDonald (1973) and later refined by Beam and Warming (1976), a spatial discretization of Eq. (3.40) can be constructed which transforms the integral conservation law for a given control volume into an ordinary differential equation (ODE) for the volume-averaged value of U . Doing this for each control volume yields a system of coupled ODEs. A separate temporal discretization step can then be applied to obtain a numerical solution at some future time level.

To begin the spatial discretization process, the surface and volume integrals in Eq. (3.40) are approximated using the following assumptions:

- The grid is fixed.
- Volume integrals are approximated as the product of the cell volume and an appropriate average value of the integrand.
- Surface integrals are approximated as the product of the vector surface area and an appropriate average value of the integrand.

Under these assumptions, the integrals in Eq. (3.38) can be approximated as

$$\frac{d}{dt} \int_{\Omega} U d\Omega \approx \frac{d}{dt} (U\Omega) \quad (3.41)$$

$$\oint_S \vec{F} \cdot \hat{n} dS \approx \sum_{l=1}^6 [\vec{F} \cdot \vec{S}]_l \quad (3.42)$$

$$\int_{\Omega} B d\Omega \approx B\Omega \quad (3.43)$$

where U , \vec{F} , and B now denote spatial averages. Introducing Eqs. (3.41) – (3.43) into Eq. (3.40) yields the ODE

$$\frac{d}{dt} (U\Omega) + \sum_{l=1}^6 [\vec{F} \cdot \vec{S}]_l = B\Omega \quad (3.44)$$

where the flux summation is given explicitly by

$$\sum_{l=1}^6 [\vec{F} \cdot \vec{S}]_l = (\vec{F} \cdot \vec{S}^\xi)_{i+\frac{1}{2}} - (\vec{F} \cdot \vec{S}^\xi)_{i-\frac{1}{2}} + (\vec{F} \cdot \vec{S}^\eta)_{j+\frac{1}{2}} - (\vec{F} \cdot \vec{S}^\eta)_{j-\frac{1}{2}} + (\vec{F} \cdot \vec{S}^\zeta)_{k+\frac{1}{2}} - (\vec{F} \cdot \vec{S}^\zeta)_{k-\frac{1}{2}} \quad (3.45)$$

Note that the flux terms employ the control volume face areas as defined by Eqs. (3.26) – (3.28). Since these area vectors point in the direction of increasing ξ , η , or

ζ , the appropriate sign must be used to account for the fact that \hat{n} in Eq. (3.40) is an outward-pointing vector.

The temporal discretization of Eq. (3.44) can be accomplished using either explicit or implicit methods. For the viscous flows considered in the present work, it may be desirable to cluster grid points in specific regions to resolve sharp gradients in velocity and density. The decrease in control volume size which results from grid clustering imposes a severe time step restriction on explicit schemes since the maximum allowable time step is directly proportional to the cell size. The present work, therefore, focuses on the use of implicit schemes.

Of the many options for implicit temporal discretization, the two-level, one step trapezoidal method is considered (Hirsh, 1988). Application of this method to Eq. (3.44) yields the following discrete equation:

$$(U^{n+1} - U^n)\Omega + \theta\Delta t \sum_{l=1}^6 [\vec{F} \cdot \vec{S} - B\Omega]_l^{n+1} + (1 - \theta)\Delta t \sum_{l=1}^6 [\vec{F} \cdot \vec{S} - B\Omega]_l^n = 0 \quad (3.46)$$

Here, n is the time index, and θ is a temporal weighting factor ranging from zero to one. Choosing $\theta = 0$ yields an explicit scheme, while choosing $\theta > 0$ results in a family of implicit schemes. Some common implicit schemes are the Crank-Nicolson scheme ($\theta = \frac{1}{2}$) and the Euler implicit scheme ($\theta = 1$).

It can be shown that the Euler implicit scheme is formally first order accurate in time, while the Crank-Nicolson scheme is second order accurate in time (Anderson, Tannehill, and Pletcher, 1984). For linear systems, both schemes are found to be unconditionally stable. For nonlinear systems, however, the Crank-Nicolson scheme can be only marginally stable (that is, some error components may grow and eventually

lead to instability). Introduction of some artificial dissipation can help minimize this problem (Hirsh, 1988).

It can be seen that Eq. (3.45) requires the six cell face areas $\vec{S}_{i\pm\frac{1}{2}}^\xi$, $\vec{S}_{j\pm\frac{1}{2}}^\eta$, and $\vec{S}_{k\pm\frac{1}{2}}^\zeta$, the cell volume Ω , and interpolated data at the cell faces (from which the fluxes can be constructed). If the flux \vec{F} contains diffusion terms, it will also be necessary to construct gradients at the cell faces. Methods for calculating these quantities are discussed in the sections below.

3.4.2 Cell face areas

Referring to the control volume shown in Fig. 3.2, it can be seen that only three of the six face areas need be associated with the grid point (i, j, k) . The convention adopted in the present work is to associate the $i + \frac{1}{2}$, $j + \frac{1}{2}$, and $k + \frac{1}{2}$ faces with grid point (i, j, k) . It follows that $\vec{S}_{i-\frac{1}{2}}^\xi$ at (i, j, k) is equivalent to $\vec{S}_{i+\frac{1}{2}}^\xi$ at $(i - 1, j, k)$, and so forth. This convention provides a convenient means of organizing the array storage of the face areas in a computer program.

Practical formulas for computing the cell face areas are based on approximate forms of Eqs. (3.26) – (3.28). Since $\Delta\xi = \Delta\eta = \Delta\zeta = 1$, the face areas can be approximated by

$$\vec{S}_{i+\frac{1}{2}}^\xi \approx \left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right)_{i+\frac{1}{2}} \Delta\eta\Delta\zeta \quad (3.47)$$

$$\vec{S}_{j+\frac{1}{2}}^\eta \approx \left(\frac{\partial \vec{r}}{\partial \zeta} \times \frac{\partial \vec{r}}{\partial \xi} \right)_{j+\frac{1}{2}} \Delta\xi\Delta\zeta \quad (3.48)$$

$$\vec{S}_{k+\frac{1}{2}}^\xi \approx \left(\overline{\frac{\partial \vec{r}}{\partial \xi}} \times \overline{\frac{\partial \vec{r}}{\partial \eta}} \right)_{k+\frac{1}{2}} \Delta \xi \Delta \eta \quad (3.49)$$

where the overbars denote appropriate mean values of the metric derivatives at the control volume face. In order to satisfy the metric identity numerically, it can be shown that the metric derivatives should be evaluated in the following manner (Vinnokur, 1989):

$$\left(\overline{\frac{\partial \vec{r}}{\partial \eta}} \right)_{i+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_a) + (\vec{r}_f - \vec{r}_e)] \quad (3.50)$$

$$\left(\overline{\frac{\partial \vec{r}}{\partial \zeta}} \right)_{i+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_f) + (\vec{r}_a - \vec{r}_e)] \quad (3.51)$$

$$\left(\overline{\frac{\partial \vec{r}}{\partial \zeta}} \right)_{j+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_f) + (\vec{r}_c - \vec{r}_g)] \quad (3.52)$$

$$\left(\overline{\frac{\partial \vec{r}}{\partial \xi}} \right)_{j+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_c) + (\vec{r}_f - \vec{r}_g)] \quad (3.53)$$

$$\left(\overline{\frac{\partial \vec{r}}{\partial \xi}} \right)_{k+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_c) + (\vec{r}_a - \vec{r}_d)] \quad (3.54)$$

$$\left(\overline{\frac{\partial \vec{r}}{\partial \eta}} \right)_{k+\frac{1}{2}} \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_a) + (\vec{r}_c - \vec{r}_d)] \quad (3.55)$$

Inserting these expressions into Eqs. (3.47) – (3.49) and simplifying yields

$$\vec{S}_{i+\frac{1}{2}}^\xi \approx \frac{1}{2} [(\vec{r}_f - \vec{r}_a) \times (\vec{r}_b - \vec{r}_e)] \quad (3.56)$$

$$\vec{S}_{j+\frac{1}{2}}^\eta \approx \frac{1}{2} [(\vec{r}_c - \vec{r}_f) \times (\vec{r}_b - \vec{r}_g)] \quad (3.57)$$

$$\vec{S}_{k+\frac{1}{2}}^\zeta \approx \frac{1}{2} [(\vec{r}_b - \vec{r}_d) \times (\vec{r}_c - \vec{r}_a)] \quad (3.58)$$

Note that point h in Fig. 3.2 is not used in computing the face areas.

3.4.3 Cell volumes

The cell volume can be calculated in a number of different ways. A relatively simple formula due to Kordulla and Vinokur (1983) approximates the volume as

$$\Omega \approx \frac{1}{3} (\vec{r}_b - \vec{r}_h) \cdot \left(\vec{S}_{i-\frac{1}{2}}^\xi + \vec{S}_{j-\frac{1}{2}}^\eta + \vec{S}_{k-\frac{1}{2}}^\zeta \right) \quad (3.59)$$

While Eq. (3.59), which arises from an analysis of various partitions of an arbitrary hexahedron, is not unique, it is efficient in that it makes use of existing data (namely the cell face areas and position vectors). Moreover, results from the literature and the author's own numerical experiments have found this formula to be reasonably accurate for smooth grids.

3.4.4 Interpolation at cell faces

This section considers the problem of interpolating a generic variable ϕ at cell face $i + \frac{1}{2}$ from data at adjacent grid points, as depicted in Fig. 3.5. Both linear and upwind-biased interpolation procedures are discussed. For simplicity, the notation is adopted wherein only indices different than i , j , or k are used in specifying a given grid point. For example, $\phi_{i,j,k}$ is written as ϕ , $\phi_{i+1,j,k}$ is written as ϕ_{i+1} , and so forth.

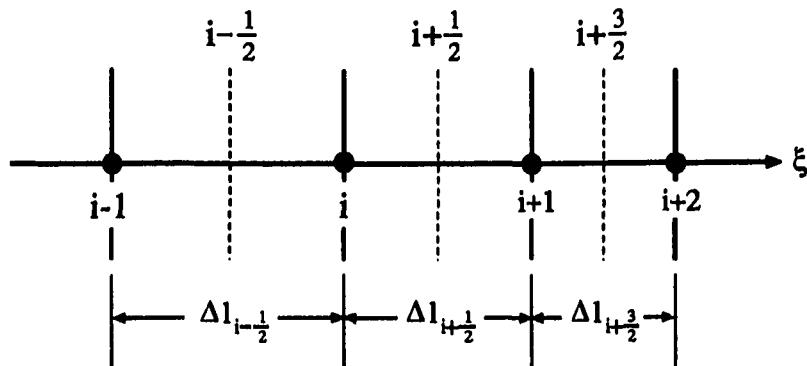


Figure 3.5: Interpolation of data at cell face $i + \frac{1}{2}$

3.4.4.1 Linear interpolation Since the control volume faces lie midway between adjacent grid points, then a linear interpolant for ϕ at cell face $i + \frac{1}{2}$ reduces to a simple arithmetic average:

$$\phi_{i+\frac{1}{2}} = \frac{1}{2} (\phi_i + \phi_{i+1}) \quad (3.60)$$

Unfortunately, the Eq. (3.60) must be modified at the boundaries due to the choice of locating grid points at boundary surfaces. To avoid the need for adding special logic to account for the boundary points in a computer program, a general interpolant which is applicable to all points can be derived as follows. Defining

$$f_{i+\frac{1}{2}}^\xi \equiv \frac{\Delta\xi}{\Delta\xi + \Delta\xi_{i+1}} \quad (3.61)$$

where, as noted earlier, $\Delta\xi$ is zero at the boundary points and one elsewhere, a linear interpolant for ϕ can be written in terms of the interpolation factor as

$$\phi_{i+\frac{1}{2}} = (1 - f_{i+\frac{1}{2}}^\xi)\phi_i + f_{i+\frac{1}{2}}^\xi\phi_{i+1} = \phi + f_{i+\frac{1}{2}}^\xi(\phi_{i+1} - \phi) \quad (3.62)$$

Similar formulas can be derived for the η and ζ directions, namely

$$\phi_{j+\frac{1}{2}} = (1 - f_{j+\frac{1}{2}}^\eta)\phi + f_{j+\frac{1}{2}}^\eta \phi_{j+1} = \phi + f_{j+\frac{1}{2}}^\eta (\phi_{j+1} - \phi) \quad (3.63)$$

$$\phi_{k+\frac{1}{2}} = (1 - f_{k+\frac{1}{2}}^\zeta)\phi + f_{k+\frac{1}{2}}^\zeta \phi_{k+1} = \phi + f_{k+\frac{1}{2}}^\zeta (\phi_{k+1} - \phi) \quad (3.64)$$

where

$$f_{j+\frac{1}{2}}^\eta \equiv \frac{\Delta\eta}{\Delta\eta + \Delta\eta_{j+1}} \quad (3.65)$$

$$f_{k+\frac{1}{2}}^\zeta \equiv \frac{\Delta\zeta}{\Delta\zeta + \Delta\zeta_{k+1}} \quad (3.66)$$

It should be noted that the interpolation factors given by Eqs. (3.61), (3.65), and (3.66) are functions of the computational coordinates only. As a result, they can be computed once and stored in three one-dimensional arrays for use as needed within the computer program.

3.4.4.2 Upwind-biased interpolation It is common in CFD to bias the interpolation of a flow variable in a direction which is “upwind” with respect to the velocity vector at a given point (Anderson, Tannehill, and Pletcher, 1984). Upwind biasing promotes both stability and accuracy of numerical solutions for convection-dominated flows. This section presents some formulas for upwind-biased interpolations which correspond to schemes of different orders of accuracy. An important feature of these formulas is their applicability to nonuniform grids.

Consider a generic flow variable ϕ which is convected in the positive ξ coordinate direction. An interpolant which corresponds to a first order upwind scheme at the $i + \frac{1}{2}$ face is given by

$$\phi_{i+\frac{1}{2}}^L = \phi \quad (3.67)$$

The superscript L is used here to denote the fact that the biasing occurs on the left hand side of the $i + \frac{1}{2}$ cell face, as shown in Fig. 3.5.

A second order upwind interpolant can be developed by extending the computational stencil to the $i - 1$ grid point (Hirsh, 1990). For convenience, a “gradient” parameter $G_{i+\frac{1}{2}}$ is first defined as

$$G_{i+\frac{1}{2}} \equiv \frac{\phi_{i+1} - \phi}{\Delta l_{i+\frac{1}{2}}^\xi} \quad (3.68)$$

The second order upwind interpolant can then be written

$$\phi_{i+\frac{1}{2}}^L = \phi + \frac{1}{2} \Delta l_{i+\frac{1}{2}}^\xi G_{i-\frac{1}{2}} \quad (3.69)$$

An interpolant corresponding to the QUICK scheme of Leonard (1979) can also be derived in terms of $G_{i+\frac{1}{2}}$ as

$$\phi_{i+\frac{1}{2}}^L = \phi + \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left[\left(\frac{2 + e_i}{1 + e_i} \right) G_{i+\frac{1}{2}} + \left(\frac{e_i}{1 + e_i} \right) G_{i-\frac{1}{2}} \right] \quad (3.70)$$

where

$$e_i \equiv \frac{\Delta l_{i+\frac{1}{2}}^\xi}{\Delta l_{i-\frac{1}{2}}^\xi} \quad (3.71)$$

If the convecting velocity is oriented in the negative ξ coordinate direction, the corresponding forms of first order, second order, and QUICK interpolants are

$$\phi_{i+\frac{1}{2}}^R = \phi \quad (3.72)$$

$$\phi_{i+\frac{1}{2}}^R = \phi - \frac{1}{2} \Delta l_{i+\frac{1}{2}}^\xi G_{i+\frac{3}{2}} \quad (3.73)$$

$$\phi_{i+\frac{1}{2}}^R = \phi_{i+1} - \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left[\left(\frac{2 + e_{i+1}}{1 + e_{i+1}} \right) G_{i+\frac{1}{2}} + \left(\frac{e_{i+1}}{1 + e_{i+1}} \right) G_{i+\frac{3}{2}} \right] \quad (3.74)$$

Here, the superscript R denotes that biasing occurs on the right hand side of the $i + \frac{1}{2}$ cell face.

The three interpolants described above can be combined into a single equation for each biasing direction. These equations can be written as

$$\phi_{i+\frac{1}{2}}^L = \phi + \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left[\epsilon_1 \left(2G_{i-\frac{1}{2}} \right) + \epsilon_2 \left(\frac{2 + e_i}{1 + e_i} \right) \left(G_{i+\frac{1}{2}} - G_{i-\frac{1}{2}} \right) \right] \quad (3.75)$$

$$\phi_{i+\frac{1}{2}}^R = \phi_{i+1} - \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left[\epsilon_1 \left(2G_{i+\frac{3}{2}} \right) + \epsilon_2 \left(\frac{1 + 2e_{i+1}}{1 + e_{i+1}} \right) \left(G_{i+\frac{1}{2}} - G_{i+\frac{3}{2}} \right) \right] \quad (3.76)$$

where various schemes are obtained by specifying appropriate values for ϵ_1 and ϵ_2 , as shown in Table 3.1.

3.4.5 Gradients at cell faces

The calculation of gradients at cell faces is based on the nonconservative form of the gradient operator. The metric terms $\nabla\xi$, $\nabla\eta$, and $\nabla\zeta$ can be written in terms of the position vector \vec{r} as follows:

Table 3.1: Upwind-biased interpolation: schemes resulting from selected values of ϵ_1 and ϵ_2

ϵ_1	ϵ_2	Scheme
0	0	first order upwind
1	0	second order upwind
1	1	QUICK

$$\nabla \xi = \frac{\left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right)}{\frac{\partial \vec{r}}{\partial \xi} \cdot \left(\frac{\partial \vec{r}}{\partial \eta} \times \frac{\partial \vec{r}}{\partial \zeta} \right)} \quad (3.77)$$

$$\nabla \eta = \frac{\left(\frac{\partial \vec{r}}{\partial \zeta} \times \frac{\partial \vec{r}}{\partial \xi} \right)}{\frac{\partial \vec{r}}{\partial \eta} \cdot \left(\frac{\partial \vec{r}}{\partial \zeta} \times \frac{\partial \vec{r}}{\partial \xi} \right)} \quad (3.78)$$

$$\nabla \zeta = \frac{\left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \eta} \right)}{\frac{\partial \vec{r}}{\partial \zeta} \cdot \left(\frac{\partial \vec{r}}{\partial \xi} \times \frac{\partial \vec{r}}{\partial \eta} \right)} \quad (3.79)$$

Thus, the gradient can be expressed exclusively in terms of derivatives of ϕ and \vec{r} with respect to the computational coordinates.

At cell faces, simple finite difference representations can be used for the derivatives. However, the forms of these derivatives will be different for each face due to the unique orientation each face has with respect to adjacent main grid points. Consequently, three sets of formulas are required (one set for each face). These formulas are presented below using the abbreviated subscript notation of Section 4.5.

- Cell Face $i + \frac{1}{2}$

$$\left(\frac{\partial \phi}{\partial \xi} \right)_{i+\frac{1}{2}} = \frac{2(\phi_{i+1} - \phi_i)}{(\Delta \xi + \Delta \xi_{i+1})} \quad (3.80)$$

$$\begin{aligned}
\left(\frac{\partial \phi}{\partial \eta} \right)_{i+\frac{1}{2}} &= \frac{2}{(\Delta \eta_{j-1} + 2\Delta \eta + \Delta \eta_{j+1})} \\
&\cdot \left\{ \frac{(\Delta \eta + \Delta \eta_{j-1})}{(\Delta \eta + \Delta \eta_{j+1})} [(1 - f^\xi)(\phi_{j+1} - \phi) + f^\xi(\phi_{i+1,j+1} - \phi_{i+1})] \right. \\
&+ \left. \frac{(\Delta \eta + \Delta \eta_{j+1})}{(\Delta \eta + \Delta \eta_{j-1})} [(1 - f^\xi)(\phi - \phi_{j-1}) + f^\xi(\phi_{i+1} - \phi_{i+1,j-1})] \right\} \\
\end{aligned} \tag{3.81}$$

$$\begin{aligned}
\left(\frac{\partial \phi}{\partial \zeta} \right)_{i+\frac{1}{2}} &= \frac{2}{(\Delta \zeta_{k-1} + 2\Delta \zeta + \Delta \zeta_{k+1})} \\
&\cdot \left\{ \frac{(\Delta \zeta + \Delta \zeta_{k-1})}{(\Delta \zeta + \Delta \zeta_{k+1})} [(1 - f^\xi)(\phi_{k+1} - \phi) + f^\xi(\phi_{i+1,k+1} - \phi_{i+1})] \right. \\
&+ \left. \frac{(\Delta \zeta + \Delta \zeta_{k+1})}{(\Delta \zeta + \Delta \zeta_{k-1})} [(1 - f^\xi)(\phi - \phi_{k-1}) + f^\xi(\phi_{i+1} - \phi_{i+1,k-1})] \right\} \\
\end{aligned} \tag{3.82}$$

- Cell Face $j + \frac{1}{2}$

$$\begin{aligned}
\left(\frac{\partial \phi}{\partial \xi} \right)_{j+\frac{1}{2}} &= \frac{2}{(\Delta \xi_{i-1} + 2\Delta \xi + \Delta \xi_{i+1})} \\
&\cdot \left\{ \frac{(\Delta \xi + \Delta \xi_{i-1})}{(\Delta \xi + \Delta \xi_{i+1})} [(1 - f^\eta)(\phi_{i+1} - \phi) + f^\eta(\phi_{i+1,j+1} - \phi_{j+1})] \right. \\
&+ \left. \frac{(\Delta \xi + \Delta \xi_{i+1})}{(\Delta \xi + \Delta \xi_{i-1})} [(1 - f^\eta)(\phi - \phi_{i-1}) + f^\eta(\phi_{j+1} - \phi_{i-1,j+1})] \right\} \\
\end{aligned} \tag{3.83}$$

$$\left(\frac{\partial \phi}{\partial \eta} \right)_{j+\frac{1}{2}} = \frac{2(\phi_{j+1} - \phi)}{(\Delta \eta + \Delta \eta_{j+1})} \quad (3.84)$$

$$\begin{aligned} \left(\frac{\partial \phi}{\partial \zeta} \right)_{j+\frac{1}{2}} &= \frac{2}{(\Delta \zeta_{k-1} + 2\Delta \zeta + \Delta \zeta_{k+1})} \\ &\cdot \left\{ \frac{(\Delta \zeta + \Delta \zeta_{k-1})}{(\Delta \zeta + \Delta \zeta_{k+1})} [(1 - f^\eta)(\phi_{k+1} - \phi) + f^\eta(\phi_{j+1,k+1} - \phi_{j+1})] \right. \\ &+ \left. \frac{(\Delta \zeta + \Delta \zeta_{k+1})}{(\Delta \zeta + \Delta \zeta_{k-1})} [(1 - f^\eta)(\phi - \phi_{k-1}) + f^\eta(\phi_{j+1} - \phi_{j+1,k-1})] \right\} \end{aligned} \quad (3.85)$$

• Cell Face $k + \frac{1}{2}$

$$\begin{aligned} \left(\frac{\partial \phi}{\partial \xi} \right)_{k+\frac{1}{2}} &= \frac{2}{(\Delta \xi_{i-1} + 2\Delta \xi + \Delta \xi_{i+1})} \\ &\cdot \left\{ \frac{(\Delta \xi + \Delta \xi_{i-1})}{(\Delta \xi + \Delta \xi_{i+1})} [(1 - f^\zeta)(\phi_{i+1} - \phi) + f^\zeta(\phi_{i+1,k+1} - \phi_{k+1})] \right. \\ &+ \left. \frac{(\Delta \xi + \Delta \xi_{i+1})}{(\Delta \xi + \Delta \xi_{i-1})} [(1 - f^\zeta)(\phi - \phi_{i-1}) + f^\zeta(\phi_{k+1} - \phi_{i-1,k+1})] \right\} \end{aligned} \quad (3.86)$$

$$\begin{aligned} \left(\frac{\partial \phi}{\partial \eta} \right)_{k+\frac{1}{2}} &= \frac{2}{(\Delta \eta_{j-1} + 2\Delta \eta + \Delta \eta_{j+1})} \\ &\cdot \left\{ \frac{(\Delta \eta + \Delta \eta_{j-1})}{(\Delta \eta + \Delta \eta_{j+1})} [(1 - f^\zeta)(\phi_{j+1} - \phi) + f^\zeta(\phi_{j+1,k+1} - \phi_{k+1})] \right. \\ &+ \left. \frac{(\Delta \eta + \Delta \eta_{j+1})}{(\Delta \eta + \Delta \eta_{j-1})} [(1 - f^\zeta)(\phi - \phi_{j-1}) + f^\zeta(\phi_{k+1} - \phi_{j-1,k+1})] \right\} \end{aligned} \quad (3.87)$$

$$\left(\frac{\partial \phi}{\partial \zeta} \right)_{k+\frac{1}{2}} = \frac{2(\phi_{k+1} - \phi)}{(\Delta \zeta + \Delta \zeta_{k+1})} \quad (3.88)$$

It should be noted that the formulas above are applicable to all points in the computational domain if the grid is specified according to the scheme presented in Section 3.2.6. In addition, they are second order accurate at interior points, but degrade to between first and second order accuracy at boundaries due to the nonuniform grid point distribution.

4. DISCRETIZATION OF THE TWO-FLUID EQUATIONS

4.1 Integral Equations

In order to apply the finite volume method described in Chapter 3 to the two-fluid system, Eqs. (2.1) – (2.3) must first be cast into an integral conservation law form which is consistent with Eq. (3.37). To this end, consider a non-material control volume Ω which is fixed relative to the (x, y, z) coordinate frame, as illustrated in Fig. 4.1. Since the free surface can move relative to the control volume, the subvolumes Ω_1 and Ω_2 and associated bounding surfaces S_1 and S_2 are functions of time. Using the relations given by Jeffrey (1965), expressions for each subvolume can be written as follows:

$$\frac{d}{dt} \int_{\Omega_1} \phi d\Omega_1 = \int_{\Omega_1} \frac{\partial \phi}{\partial t} d\Omega_1 - \int_{\Sigma} \phi \vec{W} \cdot \hat{\nu} dS \quad (4.1)$$

$$\frac{d}{dt} \int_{\Omega_2} \phi d\Omega_2 = \int_{\Omega_2} \frac{\partial \phi}{\partial t} d\Omega_2 + \int_{\Sigma} \phi \vec{W} \cdot \hat{\nu} dS \quad (4.2)$$

Adding these equations and setting $\vec{W} = \vec{V}$ in accordance with Eq. 2.15 yields

$$\frac{d}{dt} \int_{\Omega} \phi d\Omega = \int_{\Omega} \frac{\partial \phi}{\partial t} d\Omega + \int_{\Sigma} \langle \phi \rangle \vec{V} \cdot \hat{\nu} dS \quad (4.3)$$

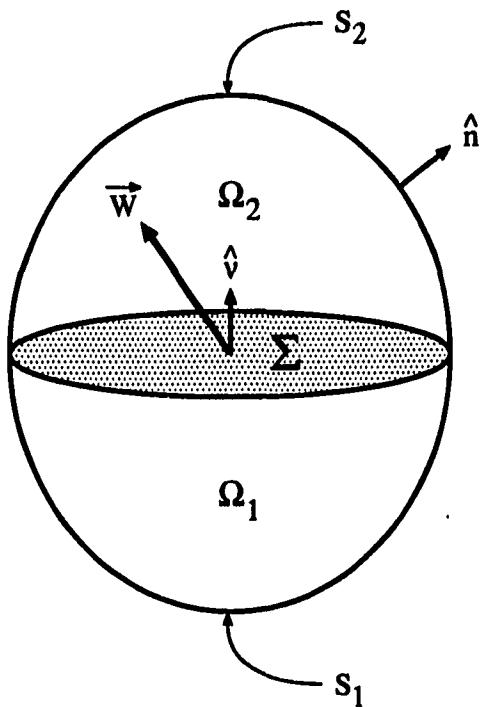


Figure 4.1: Control volume containing an interface between two immiscible fluids

Now, if Eqs. (2.1), (2.2), and (2.3) are integrated over subvolumes Ω_1 and Ω_2 , and Gauss' theorem is applied to the divergence terms, the resulting equations can be summed to give

$$\int_{\Omega} \frac{\partial \rho}{\partial t} d\Omega + \oint_S \rho \vec{V} \cdot \hat{n} dS + \int_{\Sigma} \langle \rho \rangle \vec{V} \cdot \hat{\nu} dS = 0 \quad (4.4)$$

$$\int_{\Omega} \frac{\partial \rho \vec{V}}{\partial t} d\Omega + \oint_S \rho \vec{V} (\vec{V} \cdot \hat{n} dS) + \int_{\Sigma} \langle \rho \vec{V} \rangle \vec{V} \cdot \hat{\nu} dS = \int_{\Omega} \rho \vec{B} d\Omega + \oint_S \bar{\bar{T}} \cdot \hat{n} dS + \int_{\Sigma} \langle \bar{\bar{T}} \rangle \cdot \hat{\nu} dS \quad (4.5)$$

$$\oint_S \vec{V} \cdot \hat{n} dS = 0 \quad (4.6)$$

Applying Eq. (4.3) with $\phi = \rho$ and $\phi = \rho\vec{V}$ to Eqs. (4.4) and (4.5) respectively gives

$$\frac{d}{dt} \int_{\Omega} \rho d\Omega + \oint_S \rho \vec{V} \cdot \hat{n} dS = 0 \quad (4.7)$$

$$\frac{d}{dt} \int_{\Omega} \rho \vec{V} d\Omega + \oint_S \rho \vec{V} (\vec{V} \cdot \hat{n} dS) = \int_{\Omega} \rho \vec{B} d\Omega + \oint_S \bar{\bar{\mathbf{T}}} \cdot \hat{n} dS + \int_{\Sigma} \langle \bar{\bar{\mathbf{T}}} \rangle \cdot \hat{\nu} dS \quad (4.8)$$

As demonstrated in the Section 2.4, the jump in the fluid stress tensor across the free surface due to surface tension can be modeled using Eq. (2.22). Integrating this term over Ω and substituting into Eq. (4.8) yields

$$\frac{d}{dt} \int_{\Omega} \rho \vec{V} d\Omega + \oint_S \rho \vec{V} (\vec{V} \cdot \hat{n} dS) = \int_{\Omega} \rho \vec{B} d\Omega + \oint_S \bar{\bar{\mathbf{T}}} \cdot \hat{n} dS + \int_{\Omega} \vec{F}_s d\Omega \quad (4.9)$$

Finally, Eq. (4.6) is modified following the artificial compressibility approach to give

$$\frac{d}{d\tau} \int_{\Omega} P d\Omega + \oint_S \vec{V} \cdot \hat{n} dS = 0 \quad (4.10)$$

Equations (4.7), (4.9), and (4.10) are the integral equations which will be used to develop the finite volume discretization of the two fluid system.

4.2 Column Vector Form

For convenience in developing the numerical scheme, the conservation equations for mass and momentum (Eqs. 4.7 and 4.9 above) can be written in column vector

form as follows:

$$\frac{d}{dt} \int_{\Omega} \mathbf{U} d\Omega + \oint_S \vec{\mathbf{F}} \cdot \hat{n} dS = \int_{\Omega} \mathbf{B} d\Omega \quad (4.11)$$

The variables \mathbf{U} , $\vec{\mathbf{F}}$, and \mathbf{B} are defined

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix} \quad (4.12)$$

$$\vec{\mathbf{F}} = \vec{\mathbf{F}}_i - \vec{\mathbf{F}}_v \quad (4.13)$$

$$\vec{\mathbf{F}}_i = \begin{bmatrix} \rho \vec{V} \\ \rho \vec{V} u + p \hat{i} \\ \rho \vec{V} v + p \hat{j} \\ \rho \vec{V} w + p \hat{k} \end{bmatrix} \quad (4.14)$$

$$\vec{\mathbf{F}}_v = \begin{bmatrix} 0 \\ \mu \left(\frac{\partial \vec{V}}{\partial x} + \nabla u \right) \\ \mu \left(\frac{\partial \vec{V}}{\partial y} + \nabla v \right) \\ \mu \left(\frac{\partial \vec{V}}{\partial z} + \nabla w \right) \end{bmatrix} \quad (4.15)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ B_x \\ B_y \\ B_z \end{bmatrix} \quad (4.16)$$

$$B_x = (\rho \vec{B} + \vec{F}_s) \cdot \hat{i} \quad (4.17)$$

$$B_y = (\rho \vec{B} + \vec{F}_s) \cdot \hat{j} \quad (4.18)$$

$$B_z = (\rho \vec{B} + \vec{F}_s) \cdot \hat{k} \quad (4.19)$$

It will be shown in the next section that the discretized form of Eq. (4.11) can be combined with Eq. (4.10) to form a single column vector equation for the two-fluid system.

4.3 Spatial and Temporal Discretization

The discretization approach discussed in Section 3.4.1 for a generic integral conservation law is now applied to Eq. (4.6) above and Eq. (2.26). Performing the spatial discretization first gives

$$\frac{d}{dt} (\mathbf{U}\Omega) + \sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l = \mathbf{B}\Omega \quad (4.20)$$

$$\frac{d}{d\tau} (P\Omega) + \sum_{l=1}^6 (\vec{V} \cdot \vec{S})_l = 0 \quad (4.21)$$

Using the trapezoidal scheme to advance Eq. (4.20) in physical time yields

$$\frac{\Omega}{\Delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \theta \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^{n+1} + (1 - \theta) \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^n = 0 \quad (4.22)$$

Since the pseudo-compressibility equation does not require second order accuracy in pseudo-time, the Euler implicit form of the trapezoidal scheme ($\theta = 1$) is preferred, due to its more favorable stability. Eq. (4.21) can thus be written as

$$\frac{\Omega}{\Delta\tau} (P^{m+1} - P^m) + \left[\sum_{l=1}^6 (\vec{V} \cdot \vec{S})_l \right]^{m+1} = 0 \quad (4.23)$$

where m is the pseudo-time index.

Eqs. (4.22) and (4.23) can now be combined by observing that quantities at the $m + 1$ time level must approach the solution at the next physical time level $n + 1$ in the limit as m approaches infinity. Thus, the index $n + 1$ in Eq. (4.22) can be replaced by the index $m + 1$, giving

$$\frac{\Omega}{\Delta t} (\mathbf{U}^{m+1} - \mathbf{U}^n) + \theta \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^{m+1} + (1 - \theta) \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^n = 0 \quad (4.24)$$

Equations (4.11) and (4.13) can now be written as the single column vector equation

$$\begin{aligned} & \frac{\Omega}{\Delta t} ([\mathbf{I}] - [\mathbf{I}_0]) \delta \mathbf{U} + \frac{\Omega}{\Delta t} [\mathbf{I}_0] (\delta \mathbf{U} + \mathbf{U}^m - \mathbf{U}^n) + [\mathbf{I}_\theta] \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^{m+1} \\ & + ([\mathbf{I}] - [\mathbf{I}_\theta]) \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^n = 0 \end{aligned} \quad (4.25)$$

where

$$\delta \mathbf{U} \equiv \mathbf{U}^{m+1} - \mathbf{U}^m \quad (4.26)$$

$$[\mathbf{I}] = \text{diag}(1, 1, 1, 1, 1) \quad (4.27)$$

$$[\mathbf{I}_0] = \text{diag}(1, 1, 1, 1, 0) \quad (4.28)$$

$$[\mathbf{I}_\theta] = \text{diag}(\theta, \theta, \theta, \theta, 1) \quad (4.29)$$

and \mathbf{U} , $\vec{\mathbf{F}}$, and \mathbf{B} have been redefined as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ P \end{bmatrix} \quad (4.30)$$

$$\vec{\mathbf{F}} = \vec{\mathbf{F}}_i - \vec{\mathbf{F}}_v \quad (4.31)$$

$$\vec{\mathbf{F}}_i = \begin{bmatrix} \rho \vec{V} \\ \rho \vec{V} u + \beta P i \\ \rho \vec{V} v + \beta P j \\ \rho \vec{V} w + \beta P k \\ \vec{V} \end{bmatrix} \quad (4.32)$$

$$\vec{\mathbf{F}}_v = \begin{bmatrix} 0 \\ \mu \left(\frac{\partial \vec{v}}{\partial x} + \nabla u \right) \\ \mu \left(\frac{\partial \vec{v}}{\partial y} + \nabla v \right) \\ \mu \left(\frac{\partial \vec{v}}{\partial z} + \nabla w \right) \\ 0 \end{bmatrix} \quad (4.33)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ B_x \\ B_y \\ B_z \\ 0 \end{bmatrix} \quad (4.34)$$

This formulation is similar to the implicit scheme described by Pan and Chakravarthy (1989) for incompressible flows. In the present case, however, the pseudo-time marching process is used instead of a relaxed Newton-iteration approach.

4.4 Flux Evaluation

The next step in the discretization process is to express the inviscid and viscous fluxes at cell faces in terms of \mathbf{U} at adjacent main grid points. The present work employs an upwind-biased discretization for the inviscid fluxes and a central-difference discretization for the viscous fluxes. For conciseness in the discussions below, the following notation is employed for the dot products of the inviscid and viscous flux vectors with the cell face area vectors at $i + \frac{1}{2}$:

$$\mathbf{F}_i \equiv \vec{\mathbf{F}}_i \cdot \vec{S}_{i+\frac{1}{2}}^e \quad (4.35)$$

$$\mathbf{F}_v \equiv \vec{\mathbf{F}}_v \cdot \vec{S}_{i+\frac{1}{2}}^{\epsilon} \quad (4.36)$$

The index notation follows that given previously in Section 3.4.4.

4.4.1 Inviscid flux evaluation

The formulation of the inviscid fluxes is based on the upwind-biased MUSCL scheme of van Leer (1985). This scheme, which was originally developed for the compressible Euler equations, has been applied to a wide variety of problems by various authors, including unsteady, incompressible flows (Pan and Chakravarthy, 1989).

The essential ideas behind van Leer's MUSCL scheme can be explained by considering a one-dimensional, inviscid flux on the grid shown in Fig. 4.2. A discrete flux (also called the numerical flux) is to be constructed at each cell face as part of a finite volume formulation. It is assumed that the discrete solution at each main grid point corresponds to a piecewise constant profile of \mathbf{U} within each cell. This, in turn, produces a discontinuous pair of fluid states at a given cell face. These states can be associated with left and right hand sides of the cell face, and are denoted respectively as \mathbf{U}^L and \mathbf{U}^R .

Godunov (1959) showed that a unique flux could be constructed at each cell face by solving the Riemann problem associated with the \mathbf{U}^L and \mathbf{U}^R fluid states. Once these numerical fluxes were known, a simple time integration could then be used to update the solution in each cell. Godunov's method employed an exact solution of the Riemann problem to construct the numerical flux. Due to the effort involved in computing this exact solution, later extensions of Godunov's approach employed

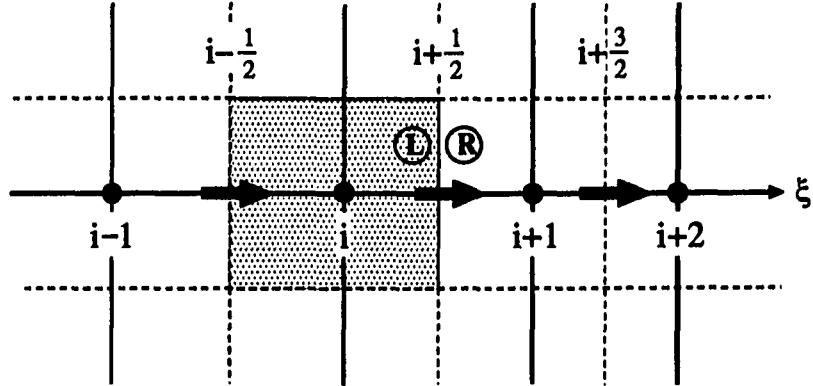


Figure 4.2: Illustration of flux evaluation for upwind-biased MUSCL scheme

approximate solutions to the Riemann problem. One of the most popular of these approximate solutions was developed by Roe (1981), and is generally referred to as Roe's scheme.

Roe's scheme is based on a local, linear wave decomposition of the inviscid flux difference across a cell face. The numerical flux arising from Roe's scheme can be expressed in the following manner. Let $[\mathbf{A}_i]$ denote the Jacobian of the inviscid flux vector \mathbf{F}_i ,

$$[\mathbf{A}_i] \equiv \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}} \quad (4.37)$$

Also, let $[\tilde{\mathbf{A}}_i]$ denote the matrix $[\mathbf{A}_i]$ evaluated at an average state $\tilde{\mathbf{U}}$, where

$$\tilde{\mathbf{U}} = \tilde{\mathbf{U}}(\mathbf{U}^L, \mathbf{U}^R) \quad (4.38)$$

Roe's numerical flux $\hat{\mathbf{F}}_i$ can then be written as

$$\hat{\mathbf{F}}_i = \frac{1}{2} [\mathbf{F}_i(\mathbf{U}^R) + \mathbf{F}_i(\mathbf{U}^L) - [\mathbf{D}_i](\mathbf{U}^R - \mathbf{U}^L)] \quad (4.39)$$

where

$$[\mathbf{D}_i] = [\tilde{\mathbf{T}}][|\tilde{\Lambda}|][\tilde{\mathbf{T}}]^{-1} \quad (4.40)$$

In the above, $[\tilde{\mathbf{T}}]$ is a matrix whose columns are the right eigenvectors of $[\tilde{\mathbf{A}}_i]$, and $[|\tilde{\Lambda}|]$ is a diagonal matrix consisting of the absolute values of the eigenvalues of $[\tilde{\mathbf{A}}_i]$. For the inviscid flux vector given by Eq. (4.32), the Jacobian matrix $[\mathbf{A}_i]$ and its associated eigenvalues and eigenvectors (and hence $[|\tilde{\Lambda}|]$, $[\tilde{\mathbf{T}}]$, and $[\tilde{\mathbf{T}}]^{-1}$) can be derived in a straightforward manner. A complete description of these matrices is provided in Appendices B and C.

Roe (1981) proposed that $\tilde{\mathbf{U}}$ be chosen based on the following constraints:

- $[\tilde{\mathbf{A}}_i]$ should have real eigenvalues and a linearly independent set of eigenvectors.
- If $\mathbf{U}^L = \mathbf{U}^R$, then

$$[\tilde{\mathbf{A}}_i] = [\mathbf{A}_i] \quad (4.41)$$

- If $\mathbf{U}^L \neq \mathbf{U}^R$, then

$$\mathbf{F}_i(\mathbf{U}^R) - \mathbf{F}_i(\mathbf{U}^L) = [\tilde{\mathbf{A}}_i] (\mathbf{U}^R - \mathbf{U}^L) \quad (4.42)$$

For the Euler equations, it can be shown that a particular average state, known as the Roe average, can be defined which satisfies the above constraints (Roe, 1981). For the two-fluid system, the appropriate Roe average is obtained by computing $\tilde{\mathbf{U}}$ as follows:

$$\tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\rho} \\ \tilde{\rho}\tilde{u} \\ \tilde{\rho}\tilde{v} \\ \tilde{\rho}\tilde{w} \\ \tilde{P} \end{bmatrix} \quad (4.43)$$

$$\tilde{\rho} = \chi \rho^L \quad (4.44)$$

$$\tilde{u} = \frac{u^L + \chi u^R}{1 + \chi} \quad (4.45)$$

$$\tilde{v} = \frac{v^L + \chi v^R}{1 + \chi} \quad (4.46)$$

$$\tilde{w} = \frac{w^L + \chi w^R}{1 + \chi} \quad (4.47)$$

$$\tilde{P} = \frac{1}{2} (P^L + P^R) \quad (4.48)$$

$$\chi = \sqrt{\frac{\rho^R}{\rho^L}} \quad (4.49)$$

The assumption of a piecewise constant profile of \mathbf{U} within each cell is equivalent to a first order upwind-biased interpolation. For example, at cell face $i + \frac{1}{2}$

$$\mathbf{U}_{i+\frac{1}{2}}^L = \mathbf{U} \quad (4.50)$$

$$\mathbf{U}_{i+\frac{1}{2}}^R = \mathbf{U}_{i+1} \quad (4.51)$$

In order to achieve a spatial accuracy higher than first order, van Leer (1985) observed that one can simply replace the piecewise constant profiles with higher order interpolations of \mathbf{U} at the cell faces. The upwind-biased MUSCL scheme uses this idea to compute the numerical flux in the following manner:

1. Define high order upwind and downwind profiles of \mathbf{U} at a given cell face.
2. Interpolate (or extrapolate) to obtain \mathbf{U}^L using the upwind profile and \mathbf{U}^R using the downwind profile.
3. Compute the numerical flux using an approximate Riemann solver (e.g. Roe's scheme).

A set of general interpolants developed by van Leer (1985) for uniform grids can be defined for cell face $i + \frac{1}{2}$ as follows:

$$\mathbf{U}_{i+\frac{1}{2}}^L = \mathbf{U} + \frac{1}{4} [(1 - \kappa)(\mathbf{U} - \mathbf{U}_{i-1}) + (1 + \kappa)(\mathbf{U}_{i+1} - \mathbf{U})] \quad (4.52)$$

$$\mathbf{U}_{i+\frac{1}{2}}^R = \mathbf{U}_{i+1} - \frac{1}{4} [(1 - \kappa)(\mathbf{U}_{i+2} - \mathbf{U}_{i+1}) + (1 + \kappa)(\mathbf{U}_{i+1} - \mathbf{U})] \quad (4.53)$$

The parameter κ in Eqs. (4.52) and (4.53) defines the order and type of interpolant. The some common schemes which result from specific choices of κ are summarized in Table 4.1.

Table 4.1: Schemes resulting from selected values of κ

κ	Scheme	Order
-1	second order upwind	2nd
0	Fromm's scheme	2nd
$\frac{1}{3}$	third order upwind	3rd
$\frac{1}{2}$	QUICK scheme	3rd
1	central difference	2nd

In the present work, the van Leer formulas are replaced with the general upwind-biased interpolants described in Section 3.4.4.2. It can be easily shown that these interpolants reduce to the corresponding van Leer formulas on a uniform grid.

An interesting interpretation of the numerical flux is observed when the van Leer interpolants are substituted into Eq. (4.39). Doing this gives

$$\hat{\mathbf{F}}_i = \frac{1}{2} [\mathbf{F}_i(\mathbf{U}^R) + \mathbf{F}_i(\mathbf{U}^L)] - \frac{1}{8} [\mathbf{D}_i](1 - \kappa)(-\mathbf{U}_{i+2} + 3\mathbf{U}_{i+1} - 3\mathbf{U} + \mathbf{U}_{i-1}) \quad (4.54)$$

The first term is seen to be a simple flux average, while the second term has the form of a discrete, third order dissipation operator. By summing fluxes of this form around a given cell, the resulting formulation is equivalent to a central difference flux plus a fourth order dissipation. Notice that the magnitude of the dissipation is controlled by κ and $[\mathbf{D}_i]$.

It is well known that higher order methods can give rise to spurious oscillations in the solution near discontinuities (Hirsch, 1990). To control these oscillations, the MUSCL scheme makes use of limiter functions. These functions essentially modify the discrete differences of U^l in the interpolation functions such that solution remains monotonic at discontinuities. For this reason, they are often called slope limiters.

Limiter functions can be incorporated into the van Leer formulas in the following

manner (Hirsch, 1990):

$$(U^l)_{i+\frac{1}{2}}^L = U^l + \frac{1}{4} \left[(1 - \kappa) (\Phi^l)_{i-\frac{1}{2}}^R (U^l - U_{i-1}^l) + (1 + \kappa) (\Phi^l)_{i+\frac{1}{2}}^L (U_{i+1}^l - U^l) \right] \quad (4.55)$$

$$(U^l)_{i+\frac{1}{2}}^R = U_{i+1}^l - \frac{1}{4} \left[(1 - \kappa) (\Phi^l)_{i+\frac{3}{2}}^L (U_{i+2}^l - U_{i+1}^l) + (1 + \kappa) (\Phi^l)_{i+\frac{1}{2}}^R (U_{i+1}^l - U^l) \right] \quad (4.56)$$

In the above, U^l denotes the l^{th} component of the solution vector \mathbf{U} , and $(\Phi^l)_{i+\frac{1}{2}}^L$ and $(\Phi^l)_{i+\frac{1}{2}}^R$ are limiter functions which are defined as

$$(\Phi^l)_{i+\frac{1}{2}}^L = \Phi \left((r^l)_{i+\frac{1}{2}}^L \right) \quad (4.57)$$

$$(\Phi^l)_{i+\frac{1}{2}}^R = \Phi \left((r^l)_{i+\frac{1}{2}}^R \right) \quad (4.58)$$

where

$$(r^l)_{i+\frac{1}{2}}^L = \frac{U^l - U_{i-1}^l + \epsilon}{U_{i+1}^l - U^l + \epsilon} \quad (4.59)$$

$$(r^l)_{i+\frac{1}{2}}^R = \frac{U_{i+2}^l - U_{i+1}^l + \epsilon}{U_{i+1}^l - U^l + \epsilon} \quad (4.60)$$

The parameter ϵ is a small number to prevent division by zero. Notice that specifying $\Phi = 1$ yields the original interpolants (i.e. no limiting is applied).

The mathematical properties of slope-limited MUSCL schemes can be developed from the theory of total variation diminishing (TVD) schemes for one dimensional,

linear conservation laws (Hirsch, 1990). From this analysis, it is found that many of the standard TVD limiters are possible candidates for the limiter functions used in Eqs. (4.55) and (4.56). Some typical limiter functions are given in Table 4.2.

Slope limiters can also be introduced into the upwind-biased interpolants of Section 3.4.4.2 by defining slope-limited forms of the gradient parameter:

$$(G^l)_{i+\frac{1}{2}}^L \equiv (\Phi^l)_{i+\frac{1}{2}}^L \left(\frac{U_{i+1}^l - U_i^l}{\Delta l_{i+\frac{1}{2}}^\xi} \right) \quad (4.61)$$

$$(G^l)_{i+\frac{1}{2}}^R \equiv (\Phi^l)_{i+\frac{1}{2}}^R \left(\frac{U_{i+1}^l - U_i^l}{\Delta l_{i+\frac{1}{2}}^\xi} \right) \quad (4.62)$$

The modified interpolants can then be written

$$(U^l)_{i+\frac{1}{2}}^L = U_i^l + \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left\{ \epsilon_1 \left[2(G^l)_{i-\frac{1}{2}}^R \right] + \epsilon_2 \left(\frac{2+e_i}{1+e_i} \right) \left[(G^l)_{i+\frac{1}{2}}^L - (G^l)_{i-\frac{1}{2}}^R \right] \right\} \quad (4.63)$$

$$(U^l)_{i+\frac{1}{2}}^R = U_{i+1}^l - \frac{1}{4} \Delta l_{i+\frac{1}{2}}^\xi \left\{ \epsilon_1 \left[2(G^l)_{i+\frac{3}{2}}^L \right] + \epsilon_2 \left(\frac{1+2e_{i+1}}{1+e_{i+1}} \right) \left[(G^l)_{i+\frac{1}{2}}^R - (G^l)_{i+\frac{3}{2}}^L \right] \right\} \quad (4.64)$$

For systems of equations, \mathbf{U}^L and \mathbf{U}^R can also be constructed by interpolating another set of variables at the cell face (such as the primitive variables or characteristic variables) and computing the corresponding conserved variables from the interpolated variables. In the present case, it is advantageous to employ the primitive variables, \mathbf{Q} , where

Table 4.2: Limiter functions applicable to the TVD MUSCL scheme

Name	Limiter Function
minmod	$\Phi(r) = \max[0, \min(1, r)]$
van Leer	$\Phi(r) = (r + r)/(1 + r)$
compressive minmod ^a	$\Phi(r) = \max[0, \min(1, \omega r)]$
superbee	$\Phi(r) = \max[0, \min(2r, 1), \min(r, 2)]$
MUSCL	$\Phi(r) = \max[0, \min(2, 2r, (1 + r)/2)]$

$$^a\omega = (3 - \kappa)/(1 - \kappa)$$

$$\mathbf{Q} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ p \end{bmatrix} \quad (4.65)$$

The primary motivation for choosing the primitive variables is that it allows a highly compressive limiter to be applied to the density to help minimize numerical diffusion at the interface. Less compressive limiters (even no limiting) can be applied separately to the velocity components and pressure. The conserved variables, on the other hand, contain the momentum components ρu , ρv , and ρw , which, because of the density, would also require a compressive limiter.

The inviscid flux construction procedure for cell face $i + \frac{1}{2}$ can now be summarized:

1. Compute $\mathbf{Q}_{i+\frac{1}{2}}^L$ and $\mathbf{Q}_{i+\frac{1}{2}}^R$ using high order, slope-limited interpolation functions (e.g. Eqs. 4.63 and 4.64).

2. Compute $\tilde{\mathbf{U}}_{i+\frac{1}{2}}^L$ and $\tilde{\mathbf{U}}_{i+\frac{1}{2}}^R$ from $\mathbf{Q}_{i+\frac{1}{2}}^L$ and $\mathbf{Q}_{i+\frac{1}{2}}^R$, and the Roe averaging relations.
3. Construct the matrices $[\tilde{\mathbf{T}}]$, $[|\tilde{\Lambda}|]$, and $[\tilde{\mathbf{T}}]^{-1}$, and thereby construct $[\mathbf{D}_i]$.
4. Compute the numerical flux from Eq. (4.39).

The one-dimensional inviscid flux formulas developed above can be employed in two and three-dimensional formulations by simply using analogous formulas for each coordinate direction. That is, the i index is simply replaced by the j and k indices in order to obtain the numerical flux in the η and ζ directions, respectively. This practice, however, assumes that information travels along waves which propagate in directions normal to the cell faces. While this assumption is clearly violated if the local velocity vectors are not oriented normal to a given cell face, good results have nonetheless been observed in practice, especially for high order schemes (Roe, 1986).

4.4.2 Viscous flux evaluation

Because the viscous fluxes are diffusive, and hence have an elliptic character, it is appropriate to use central difference discretizations in their construction. In the present case, the viscous flux vector $\vec{\mathbf{F}}_v$ is constructed directly at a given cell face. For example, at cell face $i + \frac{1}{2}$,

$$(\vec{\mathbf{F}}_v)_{i+\frac{1}{2}} = \begin{bmatrix} 0 \\ \mu_{i+\frac{1}{2}} \left(\frac{\partial \vec{v}}{\partial x} + \nabla u \right)_{i+\frac{1}{2}} \\ \mu_{i+\frac{1}{2}} \left(\frac{\partial \vec{v}}{\partial y} + \nabla v \right)_{i+\frac{1}{2}} \\ \mu_{i+\frac{1}{2}} \left(\frac{\partial \vec{v}}{\partial z} + \nabla w \right)_{i+\frac{1}{2}} \\ 0 \end{bmatrix} \quad (4.66)$$

The viscosity required in the above is computed by linearly interpolating from neighbor main grid point values,

$$\mu_{i+\frac{1}{2}} = \frac{1}{2} (\mu + \mu_{i+1}) \quad (4.67)$$

It should be noted that since the viscosity is uniform within each fluid, the numerical value of μ at a given point (i, j, k) can be derived from knowledge of the density distribution as follows. First, define a parameter α as

$$\alpha \equiv \frac{\rho - \rho_2}{\rho_1 - \rho_2} \quad (4.68)$$

where ρ_1 and ρ_2 are the densities of primary and secondary fluids, respectively. It can be seen that α is one in the liquid and zero in the gas. Accordingly, the viscosity can be computed from

$$\mu = \alpha \mu_1 + (1 - \alpha) \mu_2 \quad (4.69)$$

where μ_1 and μ_2 are the dynamic viscosities of the primary and secondary fluids, respectively.

The derivatives $\partial \vec{V} / \partial x$, $\partial \vec{V} / \partial y$, and $\partial \vec{V} / \partial z$ can be determined from ∇u , ∇v , and ∇w using

$$\frac{\partial \vec{V}}{\partial x} = (\nabla u \cdot \hat{i}) \hat{i} + (\nabla v \cdot \hat{i}) \hat{j} + (\nabla w \cdot \hat{i}) \hat{k} \quad (4.70)$$

$$\frac{\partial \vec{V}}{\partial y} = (\nabla u \cdot \hat{j}) \hat{i} + (\nabla v \cdot \hat{j}) \hat{j} + (\nabla w \cdot \hat{j}) \hat{k} \quad (4.71)$$

$$\frac{\partial \vec{V}}{\partial z} = (\nabla u \cdot \hat{k})\hat{i} + (\nabla v \cdot \hat{k})\hat{j} + (\nabla w \cdot \hat{k})\hat{k} \quad (4.72)$$

Thus, only ∇u , ∇v , and ∇w and μ are needed at the cell face in order to construct the entire viscous flux vector. The gradients can be easily computed using the formulas presented in Section 3.4.5.

4.5 Linearization

The final step of the discretization process is to linearize the $m+1$ pseudo-time level terms about pseudo-time level m . Doing this for each control volume yields a block-banded system of linear equations which can be solved using the methods discussed in the next chapter.

To begin, the total flux \mathbf{F}^{m+1} is expanded in a Taylor series about m as follows:

$$\mathbf{F}^{m+1} = \mathbf{F}^m + \left(\frac{\partial \mathbf{F}}{\partial \tau} \right)^m \Delta \tau + \mathcal{O}[(\Delta \tau)^2] \quad (4.73)$$

where $\mathcal{O}[(\Delta \tau)^2]$ denotes terms of order $(\Delta \tau)^2$ and higher. Following Beam and Warming (1976), the higher order terms are neglected and a discrete approximation for the first derivative term is constructed. Before doing this, the flux approximation is split into inviscid and viscous parts:

$$\mathbf{F}^{m+1} = \mathbf{F}_i^{m+1} + \mathbf{F}_v^{m+1} \quad (4.74)$$

$$\mathbf{F}_i^{m+1} = \mathbf{F}_i^m + \left(\frac{\partial \mathbf{F}_i}{\partial \tau} \right)^m \Delta \tau \quad (4.75)$$

$$\mathbf{F}_v^{m+1} = \mathbf{F}_v^m + \left(\frac{\partial \mathbf{F}_v}{\partial \tau} \right)^m \Delta \tau \quad (4.76)$$

The inviscid flux time derivative term in Eq. (4.75) is expanded using the chain rule

$$\left(\frac{\partial \mathbf{F}_i}{\partial \tau} \right)^m = \left(\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}} \right)^m \frac{\partial \mathbf{U}}{\partial \tau} \quad (4.77)$$

Using the numerical flux $\hat{\mathbf{F}}_i = \mathbf{F}_i(\mathbf{U}^L, \mathbf{U}^R)$ in this equation gives

$$\left(\frac{\partial \mathbf{F}_i}{\partial \tau} \right)^m = \left(\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^L} \right)^m \frac{\partial \mathbf{U}^L}{\partial \tau} + \left(\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^R} \right)^m \frac{\partial \mathbf{U}^R}{\partial \tau} \quad (4.78)$$

The pseudo-time derivatives of \mathbf{U}^L and \mathbf{U}^R are now approximated with finite differences:

$$\frac{\partial \mathbf{U}^L}{\partial \tau} \approx \frac{(\mathbf{U}^L)^{m+1} - (\mathbf{U}^L)^m}{\Delta \tau} = \frac{\delta \mathbf{U}^L}{\Delta \tau} \quad (4.79)$$

$$\frac{\partial \mathbf{U}^R}{\partial \tau} \approx \frac{(\mathbf{U}^R)^{m+1} - (\mathbf{U}^R)^m}{\Delta \tau} = \frac{\delta \mathbf{U}^R}{\Delta \tau} \quad (4.80)$$

Inserting these approximations into Eq. (4.78) yields

$$\left(\frac{\partial \mathbf{F}_i}{\partial \tau} \right)^m \approx \left(\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^L} \right)^m \frac{\delta \mathbf{U}^L}{\Delta \tau} + \left(\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^R} \right)^m \frac{\delta \mathbf{U}^R}{\Delta \tau} \quad (4.81)$$

It should be noted that both $\delta \mathbf{U}^L$ and $\delta \mathbf{U}^R$ can be expressed in terms of $\delta \mathbf{U}$ at the main grid points using the interpolation functions described in the previous section.

The Jacobians of the numerical flux $\partial \hat{\mathbf{F}}_i / \partial \mathbf{U}^L$ and $\partial \hat{\mathbf{F}}_i / \partial \mathbf{U}^R$ can be derived from the definition of Roe's numerical flux, Eq. (4.39). Following Barth (1987), the

Jacobians are simplified by neglecting terms arising from the derivatives of $[\mathbf{D}_i]$ with respect to \mathbf{U}^L and \mathbf{U}^R . The resulting simplified Jacobians can be written as

$$\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^L} \approx \frac{1}{2} \left[[\mathbf{A}_i(\hat{\mathbf{U}})] + [\mathbf{D}_i] \right] \equiv [\mathbf{A}_i^+] \quad (4.82)$$

$$\frac{\partial \hat{\mathbf{F}}_i}{\partial \mathbf{U}^R} \approx \frac{1}{2} \left[[\mathbf{A}_i(\mathbf{U}^R)] - [\mathbf{D}_i] \right] \equiv [\mathbf{A}_i^-] \quad (4.83)$$

Substituting the simplified Jacobians in Eq. (4.78) gives

$$\left(\frac{\partial \mathbf{F}_i}{\partial \tau} \right)^m \approx \frac{1}{\Delta \tau} \left[[\mathbf{A}_i^+] \delta \mathbf{U}^L + [\mathbf{A}_i^-] \delta \mathbf{U}^R \right] \quad (4.84)$$

Thus, the linearized inviscid flux becomes

$$\mathbf{F}_i^{m+1} \approx \mathbf{F}_i^m + [\mathbf{A}_i^+] \delta \mathbf{U}^L + [\mathbf{A}_i^-] \delta \mathbf{U}^R \quad (4.85)$$

For compactness, a flux difference $\delta \mathbf{F}_i$ is defined by

$$\delta \mathbf{F}_i \equiv [\mathbf{A}_i^+] \delta \mathbf{U}^L + [\mathbf{A}_i^-] \delta \mathbf{U}^R \quad (4.86)$$

so that Eq. (4.85) can be written

$$\mathbf{F}_i^{m+1} \approx \mathbf{F}_i^m + \delta \mathbf{F}_i \quad (4.87)$$

The linearization of the viscous can be carried out in a similar fashion, yielding an expression similar to Eq. (4.87) above:

$$\mathbf{F}_v^{m+1} \approx \mathbf{F}_v^m + \delta \mathbf{F}_v \quad (4.88)$$

However, the Jacobian matrices which arise in $\delta\mathbf{F}_v$, will be relatively complex if derived in full. As a result, it is common to either employ a simplified form of $\delta\mathbf{F}_v$ (see, for example, Rogers, Kwak, and Kiris, 1989) or to neglect it altogether. In the present work, the following simplified form of $\delta\mathbf{F}_v$ (derived in Appendix D) can be employed at cell face $i + \frac{1}{2}$:

$$\delta\mathbf{F}_v = [\mathbf{D}_v](\delta\mathbf{U}_{i+1} - \delta\mathbf{U}) \quad (4.89)$$

This expression neglects non-orthogonal terms in the gradients, and therefore can be expected to be a poor approximation on highly skewed, non-orthogonal grids.

The linearization of the body force term is reasonably straightforward, the result being represented as

$$\mathbf{B}^{m+1} \approx \mathbf{B}^m + [\mathbf{G}]\delta\mathbf{U} \quad (4.90)$$

where

$$[\mathbf{G}] \equiv \frac{\partial \mathbf{B}}{\partial \mathbf{U}} \quad (4.91)$$

A form of the matrix $[\mathbf{G}]$ applicable to the present work is derived in Appendix E.

The foregoing linearizations can now be introduced into Eq. (4.25), giving

$$\begin{aligned} & \frac{\Omega}{\Delta\tau} ([\mathbf{I}] - [\mathbf{I}_0]) \delta\mathbf{U} + \frac{\Omega}{\Delta t} [\mathbf{I}_0] (\delta\mathbf{U} + \mathbf{U}^m - \mathbf{U}^n) + [\mathbf{I}_\theta] \left[\sum_{l=1}^6 (\delta\mathbf{F}_l - \delta\mathbf{F}_v)_l - [\mathbf{G}]\Omega \right]^m \delta\mathbf{U} \\ & + [\mathbf{I}_\theta] \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^m + ([\mathbf{I}] - [\mathbf{I}_\theta]) \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^n = 0 \end{aligned} \quad (4.92)$$

To facilitate the development of the solution algorithm, all terms not associated with $\delta\mathbf{U}$ are placed on the right hand side of Eq. (4.92). The final form of the discretized equation can then be written as

$$\frac{\Omega}{\Delta\tau} ([\mathbf{I}] + [\mathbf{H}]) \delta\mathbf{U} + [I_\theta] \left[\sum_{l=1}^6 (\delta\mathbf{F}_i - \delta\mathbf{F}_v)_l \right]^m = -\mathbf{R} \quad (4.93)$$

where

$$\mathbf{R} = \frac{\Omega}{\Delta t} [\mathbf{I}_0] (\mathbf{U}^m - \mathbf{U}^n) + [\mathbf{I}_\theta] \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^m + ([\mathbf{I}] - [\mathbf{I}_\theta]) \left[\sum_{l=1}^6 (\vec{\mathbf{F}} \cdot \vec{S})_l - \mathbf{B}\Omega \right]^n \quad (4.94)$$

$$[\mathbf{H}] = \left(\frac{\Delta\tau}{\Delta t} - 1 \right) [\mathbf{I}_0] - \Delta\tau [\mathbf{I}_\theta] [\mathbf{G}]^m \quad (4.95)$$

Notice that setting \mathbf{R} equal to zero yields a second order accurate discretization of Eq. (4.25) in both space and time for $\theta = \frac{1}{2}$. Therefore, as long as the solution converges in pseudo-time (i.e. $\delta\mathbf{U} \rightarrow 0$), the computed solution is unaffected by the approximations made on the left hand side.

Applying Eq. (4.93) to all control volumes and introducing appropriate expressions for $\delta\mathbf{F}_i$ and $\delta\mathbf{F}_v$ yields a block system of algebraic equations for the unknowns $\delta\mathbf{U}$. The solution at the next time level is obtained by marching the solution in pseudo-time until a steady state condition is achieved. In order to promote the stability of the pseudo-time marching process, it is desirable to enhance the diagonal dominance of the system of equations. One way of doing this in the present case is to add pseudo-time derivative terms to the mass and momentum equations. Note that this does not affect the computed solution since all pseudo-time derivatives should

approach zero in the limit as $m \rightarrow \infty$. Adding pseudo-time derivatives to Eq. (4.93) results in the following modified form of the $[\mathbf{H}]$ matrix:

$$[\mathbf{H}] = \frac{\Delta\tau}{\Delta t} [\mathbf{I}_0] - \Delta\tau [\mathbf{I}_\theta] [\mathbf{G}] \quad (4.96)$$

Another method for enhancing diagonal dominance is to employ a local pseudo-time step for each cell. An expression for the local pseudo-time step can be derived using the maximum eigenvalue (or spectral radius) of the flux Jacobian (Hirsch, 1990). Denoting the spectral radii associated with the three computational coordinate directions as σ^ξ , σ^η , and σ^ζ , the local pseudo-time step can be determined from

$$\Delta\tau = \text{CFL} \min\left(\frac{\Omega}{\sigma^\xi}, \frac{\Omega}{\sigma^\eta}, \frac{\Omega}{\sigma^\zeta}\right) \quad (4.97)$$

where CFL is a constant on the order of one. For the flow problems considered in the present work, it was found sufficient to use the spectral radii of the inviscid flux Jacobians alone (see Appendix C).

4.6 Boundary Conditions

Boundary conditions are incorporated into the numerical formulation by expressing $\delta\mathbf{U}$ at boundary grid points in terms of both user-prescribed boundary data and the solution at interior grid points. For example, at the constant ξ boundary $i = 1$, a boundary equation can be written as

$$\delta\mathbf{U} = [\mathbf{C}_{w1}^b]\delta\mathbf{U}_{i+1} + [\mathbf{C}_{w2}^b]\delta\mathbf{U}_{i+2} + \mathbf{C}_{w3}^b \quad (4.98)$$

where $[\mathbf{C}_{w1}^b]$, $[\mathbf{C}_{w2}^b]$, and \mathbf{C}_{w3}^b are boundary coefficients. If this equation is substituted

into Eq. (4.92) for grid point $i = 2$ (with $\delta\mathbf{F}_i$ and $\delta\mathbf{F}_v$ expressed in terms of $\delta\mathbf{U}$), then all references to the solution at $i = 1$ are eliminated. Applying this technique to all boundary points produces a block system of equations for unknowns defined for $i = 2$ to $i_{max} - 1$, $j = 2$ to $j_{max} - 1$, and $k = 2$ to $k_{max} - 1$.

In order to derive the particular boundary coefficients for use in the present formulation, the boundary conditions defined in Section 2.6 must first be cast in terms of the solution vector \mathbf{U} . The resulting expressions may then be linearized to obtain equations of the form given by Eq. (4.98). To illustrate this procedure, the boundary equations for the boundary surface $i = 1$ are derived below. Note that U^1, U^2, \dots denote the components of the solution vector \mathbf{U} .

- No Slip

The no slip condition implies that

$$U^2 = U^3 = U^4 = 0 \quad (4.99)$$

Linearizing Eq. (4.99) about time level $m + 1$ gives

$$(U^2)^{m+1} = (U^2)^m + \delta U^2 \quad (4.100)$$

$$(U^3)^{m+1} = (U^3)^m + \delta U^3 \quad (4.101)$$

$$(U^4)^{m+1} = (U^4)^m + \delta U^4 \quad (4.102)$$

If Eq. (4.99) is imposed in the initial conditions, then the above relations reduce to

$$\delta U^2 = 0 \quad (4.103)$$

$$\delta U^3 = 0 \quad (4.104)$$

$$\delta U^4 = 0 \quad (4.105)$$

- Pressure

From Section 2.6, the simplified boundary condition for pressure is given by

$$\frac{\partial p}{\partial n} \approx \rho B_n \quad (4.106)$$

where, in this case, n denotes the normal direction with respect to the constant ξ boundary surface at $i = 1$. It will be assumed here that the grid is locally orthogonal at the wall, so that $\partial p / \partial n$ can be expressed as

$$\frac{\partial p}{\partial n} = |\nabla \xi| \frac{\partial p}{\partial \xi} \quad (4.107)$$

Using the first order approximations

$$\frac{\partial p}{\partial \xi} \approx \frac{p_{i+1} - p_i}{\xi_{i+1} - \xi_i} \quad (4.108)$$

$$|\nabla \xi| \approx \frac{|\vec{r}_{i+1} - \vec{r}_i|}{\xi_{i+1} - \xi_i} \quad (4.109)$$

$$\hat{n} \approx \frac{\vec{r}_{i+1} - \vec{r}}{|\vec{r}_{i+1} - \vec{r}|} \quad (4.110)$$

the discrete boundary equation for pressure can be written as

$$p_{i+1} - p = \rho \vec{B} \cdot (\vec{r}_{i+1} - \vec{r}) \quad (4.111)$$

Dividing Eq. (4.110) by β and linearizing yields

$$\delta U^5 = \delta U_{i+1}^5 + \left[U_{i+1}^5 - U^5 + \frac{\rho \vec{B}}{\beta} \cdot (\vec{r}_{i+1} - \vec{r}) \right] \quad (4.112)$$

Note that the body force term $\rho \vec{B}$ is evaluated at the wall.

- Density

The boundary value of density is derived from a simple zeroth order extrapolation:

$$U^1 = U_{i+1}^1 \quad (4.113)$$

Linearizing about time level $m + 1$ gives

$$(U^1)^m + \delta U^1 = (U_{i+1}^1)^m + \delta U_{i+1}^1 \quad (4.114)$$

Rearranging this equation yields

$$\delta U^1 = \delta U_{i+1}^1 + (U_{i+1}^1 - U^1)^m \quad (4.115)$$

4.7 Nondimensional Equations

Using the nondimensionalization procedure outlined in Section 2.8, the discretized system of equations may now be cast into the following form:

$$\frac{\Omega^*}{\Delta\tau^*} ([\mathbf{I}] + [\mathbf{H}]) \delta \mathbf{U} + [\mathbf{I}_\theta] \left[\sum_{i=1}^6 (\delta \mathbf{F}_i - \delta \mathbf{F}_v)_i \right]^m = -\mathbf{R} \quad (4.116)$$

$$\mathbf{R} = \frac{\Omega^*}{\Delta t^*} [\mathbf{I}_0] (\mathbf{U}^m - \mathbf{U}^n) + [\mathbf{I}_\theta] \left[\sum_{i=1}^6 (\vec{\mathbf{F}} \cdot \vec{S}^*)_i - \mathbf{B} \Omega^* \right]^m + ([\mathbf{I}] - [\mathbf{I}_\theta]) \left[\sum_{i=1}^6 (\vec{\mathbf{F}} \cdot \vec{S}^*)_i - \mathbf{B} \Omega^* \right]^n \quad (4.117)$$

$$[\mathbf{H}] = \left(\frac{\Delta\tau^*}{\Delta t^*} - 1 \right) [\mathbf{I}_0] - \Delta\tau^* [\mathbf{I}_\theta] [\mathbf{G}]^m \quad (4.118)$$

\mathbf{U} , $\vec{\mathbf{F}}$, and \mathbf{B} are now defined as

$$\mathbf{U} = \begin{bmatrix} \rho^* \\ \rho^* u^* \\ \rho^* v^* \\ \rho^* w^* \\ P \end{bmatrix} \quad (4.119)$$

$$\vec{\mathbf{F}} = \vec{\mathbf{F}}_i - \vec{\mathbf{F}}_v \quad (4.120)$$

$$\vec{\mathbf{F}}_i = \begin{bmatrix} \rho^* \vec{V}^* \\ \rho^* \vec{V}^* u^* + \beta^* P \hat{i} \\ \rho^* \vec{V}^* v^* + \beta^* P \hat{j} \\ \rho^* \vec{V}^* w^* + \beta^* P \hat{k} \\ \vec{V}^* \end{bmatrix} \quad (4.121)$$

$$\vec{\mathbf{F}}_v = \begin{bmatrix} 0 \\ \frac{\mu^*}{Re} \left(\frac{\partial \vec{V}^*}{\partial x^*} + \nabla u^* \right) \\ \frac{\mu^*}{Re} \left(\frac{\partial \vec{V}^*}{\partial y^*} + \nabla v^* \right) \\ \frac{\mu^*}{Re} \left(\frac{\partial \vec{V}^*}{\partial z^*} + \nabla w^* \right) \\ 0 \end{bmatrix} \quad (4.122)$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ B_x^* \\ B_y^* \\ B_z^* \\ 0 \end{bmatrix} \quad (4.123)$$

5. SOLUTION ALGORITHM

5.1 Implicit Time Marching Algorithm

Writing Eq. (4.93) for all control volumes and incorporating boundary equations of the form given by Eq. (4.98) yields a large system of linear, algebraic equations. This system can be written symbolically as

$$[\mathcal{M}]\{\delta\mathbf{U}\} = -\{\mathbf{R}\} \quad (5.1)$$

where $[\mathcal{M}]$ is an $N_{cv} \times N_{cv}$ block-banded matrix, and $\{\delta\mathbf{U}\}$ and $\{\mathbf{R}\}$ are vectors of length N_{cv} .

The basic solution algorithm consists of solving Eq. (5.1) for $\{\delta\mathbf{U}\}$, updating the current solution, and repeating this iterative process until a steady state solution in pseudo-time is obtained. This steady state solution then becomes the solution at the next physical time level, \mathbf{U}^{n+1} . This approach is very similar to a Newton-iteration algorithm with the pseudo-time acting as a relaxation mechanism. Algorithms of this type have been explored by various authors, including Merkle and Athavale (1987), Rogers, Kwak, and Kiris (1989), and Pan and Chakravarthy (1989).

A more detailed definition of the basic implicit time marching algorithm can be written as follows:

```

prescribe initial data,  $\mathbf{U}^m$ .
do  $n = 1$  to  $n_{max}$ 
  prescribe physical time step,  $\Delta t$ .
  set  $\{\mathbf{U}^n\} = \{\mathbf{U}^m\}$ .
  do  $m = 1$  to  $m_{max}$ 
    compute pseudo-time step,  $\Delta\tau$ .
    compute  $\{\mathbf{R}\}$ .
    compute  $\{\mathcal{M}\}$ .
    solve Eq. (5.1) for  $\{\delta\mathbf{U}\}$ .
    update solution:  $\{\mathbf{U}^m\} = \{\mathbf{U}^m\} + \{\delta\mathbf{U}^m\}$ .
    if solution has converged, then exit do loop.
  end do
end do

```

It is clear that the efficiency of this algorithm depends on two factors: the computational effort required to solve Eq. (5.1), and the convergence rate of the pseudo-time marching process. Unfortunately, a solution algorithm which is computationally inexpensive tends to converge poorly, while an algorithm which converges rapidly tends to be computationally expensive. An optimal approach will therefore involve some compromise between the competing goals of minimal computational effort and rapid convergence.

Another issue which is directly related to algorithm performance is the convergence criteria used in the psuedo-time marching loop. Ideally, one would like to drive the residual vector $\{\mathbf{R}\}$ to machine zero at each physical time step. This goal, however, can only rarely be achieved in practice, since it would require a large number of

subiterations. A more practical requirement is to halt the subiterations when some measure of the residual is smaller than a prescribed tolerance. In the present work, the L_2 norm of the components of the residual are used for this purpose:

$$\mathcal{R}^l \equiv \sqrt{\frac{1}{N_{cv}} \sum_{N_{cv}} (R^l)^2} \quad (l = 1, 2, 3, 4, 5) \quad (5.2)$$

The pseudo-time marching solution is deemed to be converged when $\mathcal{R}^l < \epsilon_r^l$ for all l , where ϵ_r^l is the convergence tolerance.

5.2 Solution Methods for Block-Banded Systems

Most methods for solving block-banded systems of equations can be divided into two categories: unfactored schemes, and approximately factored schemes. Unfactored schemes attempt to solve Eq. (5.1) as it stands, and usually employ a block version of a standard iterative method for linear equations. Some examples of unfactored schemes include the line Gauss-Seidel relaxation method (LGS), the conjugate gradient method (CG), and the strongly implicit method (SIP). Approximately factored schemes, on the other hand, approximate $[\mathcal{M}]$ as the product of two or more factors which are relatively easy to invert. Some examples of factored schemes include the Beam-Warming alternating direction implicit scheme (ADI) and the LU factored scheme (LU).

Approximately factored schemes are typically faster (per iteration) and have lower memory requirements than unfactored schemes. However, the error introduced by the approximate factorization of $[\mathcal{M}]$ limits the effective time step which can be taken and hence reduces the convergence rate. Consequently, the total computational effort required to obtain a converged solution may be comparable for both approaches.

In fact, evidence from the literature suggests that some unfactored schemes can be significantly more efficient than factored schemes for many problems.

The choice of which methods to use in the present work was determined after developing computer programs based on several approximately factored and unfactored schemes, and evaluating their performance on a standard model problem (three-dimensional, incompressible flow in a driven, cubic cavity). The discretization used in these programs was based on a finite volume version of the unsteady, pseudo-compressible algorithm of Rogers, Kwak, and Kiris (1989). The solution methods evaluated included the LGS, ADI, and LU schemes.

From these numerical tests, it was found that while all three methods could be used successfully to compute three-dimensional, unsteady flows, the LU scheme required the least total computational effort and possessed lower computer memory requirements as compared to the LGS and ADI schemes. In addition, Yoon, Kwak, and Chang (1989) have shown that the LU scheme can be completely vectorized, making it a good choice for use on supercomputers. Based on these results, the LU scheme was adopted as the baseline solver for the surface capturing algorithm. The particular form of the LU scheme used in the present work will be described in Section 5.4.

For geometries with periodic (or reentrant) boundaries, such as a three-cylindrical or spherical tank, a modified version of the LU algorithm can be devised which properly handles the additional coefficients which arise due to periodic boundaries (Buratynski and Caughey, 1986). Unfortunately, the modified algorithm is significantly more complex and requires about twice the computational effort as compared to the non-periodic LU scheme. Upon further study, it was determined that an LGS

scheme could be implemented much more easily for the periodic geometries considered in the present work with only a modest increase in computational effort relative to the periodic LU scheme. In particular, a periodic tridiagonal solver could be applied in the coordinate direction containing the periodic boundary, while a non-periodic tridiagonal solver could be applied in the other two coordinate directions. Additional details of this LGS scheme are provided in Section 5.5.

5.3 Final Form of the Discretized Equations

From the definition of the inviscid numerical flux (Eq. 4.38), it is seen that the use of a high order interpolant for $\delta\mathbf{U}^L$ and $\delta\mathbf{U}^R$ results in a large bandwidth matrix operator. To reduce this bandwidth, it is common to employ first order upwind interpolation for the left hand side operator. Thus, the linearized inviscid flux terms use the following:

$$\mathbf{U}_{i+\frac{1}{2}}^L = \mathbf{U} \quad (5.3)$$

$$\mathbf{U}_{i+\frac{1}{2}}^R = \mathbf{U}_{i+1} \quad (5.4)$$

$$\delta\mathbf{U}_{i+\frac{1}{2}}^L = \delta\mathbf{U} \quad (5.5)$$

$$\delta\mathbf{U}_{i+\frac{1}{2}}^R = \delta\mathbf{U}_{i+1} \quad (5.6)$$

$$[\mathbf{A}_i^+]_{i+\frac{1}{2}} = \frac{1}{2} ([\mathbf{A}_i(\mathbf{U})] + [\tilde{\mathbf{D}}])_{i+\frac{1}{2}} \quad (5.7)$$

$$[\mathbf{A}_i^-]_{i+\frac{1}{2}} = \frac{1}{2} ([\mathbf{A}_i(\mathbf{U}_{i+1})] - [\tilde{\mathbf{D}}])_{i+\frac{1}{2}} \quad (5.8)$$

Notice that the first order upwind assumption is equivalent to neglecting the second term in the high order interpolation functions (Eqs. 4.63 and 4.64).

Using this simplification of the left hand side operator, Eq. (4.92) can now be written

$$\begin{aligned} & \frac{\Omega}{\Delta\tau} ([\mathbf{I}] + [\mathbf{H}]) \delta\mathbf{U} + [\mathbf{I}_\theta] \left\{ \left[([\mathbf{A}_i^+] + [\mathbf{D}_v])_{i+\frac{1}{2}} - ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{i-\frac{1}{2}} + ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{j+\frac{1}{2}} \right. \right. \\ & \quad \left. \left. - ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{j-\frac{1}{2}} + ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{k+\frac{1}{2}} - ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{k-\frac{1}{2}} \right] \delta\mathbf{U} \right. \\ & \quad \left. + ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{i+\frac{1}{2}} \delta\mathbf{U}_{i+1} - ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{i-\frac{1}{2}} \delta\mathbf{U}_{i-1} + ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{j+\frac{1}{2}} \delta\mathbf{U}_{j+1} \right. \\ & \quad \left. - ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{j-\frac{1}{2}} \delta\mathbf{U}_{j-1} + ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{k+\frac{1}{2}} \delta\mathbf{U}_{k+1} - ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{k-\frac{1}{2}} \delta\mathbf{U}_{k-1} \right\} \\ & \quad = -\mathbf{R} \end{aligned} \quad (5.9)$$

By introducing boundary equations of the form given by Eq. (4.98), the discrete, linearized system can be cast into the following form:

$$\begin{aligned} & \frac{\Omega}{\Delta\tau} ([\mathbf{I}] + [\mathbf{H}]) \delta\mathbf{U} + [\mathbf{C}_P] \delta\mathbf{U} + [\mathbf{C}_E] \delta\mathbf{U}_{i+1} + [\mathbf{C}_W] \delta\mathbf{U}_{i-1} + [\mathbf{C}_N] \delta\mathbf{U}_{j+1} \\ & \quad + [\mathbf{C}_S] \delta\mathbf{U}_{j-1} + [\mathbf{C}_F] \delta\mathbf{U}_{k+1} + [\mathbf{C}_B] \delta\mathbf{U}_{k-1} = -\mathbf{R}' \end{aligned} \quad (5.10)$$

The coefficient matrices and modified right hand side vector are defined in detail in Appendix F.

It can be seen that equations of the form given by Eq. (5.10) provide for a fully implicit treatment of boundary conditions. Therefore, once the boundary equations

coefficients have been specified, a block system of equations is completely defined for all interior main grid points.

5.4 The LU Scheme

The LU scheme was originally proposed by Jameson and Turkel (1981) as a means of constructing well-conditioned factorizations of implicit schemes for hyperbolic equations. It has since been applied to the solution of the Euler equations by Burakynski and Caughey (1986), Jameson and Yoon (1987), and Yakota and Caughey (1988), and to the solution of the Navier-Stokes equations by Pan and Lomax (1988) and Yakota (1990), among others.

The LU scheme approximately factors the block-banded matrix operator $[M]$ into the product of a lower diagonal operator $[L]$ and an upper diagonal operator $[U]$. Neglecting the second order factorization error, the modified system can be written as

$$[L][U]\{\delta U\} = -\{R\} \quad (5.11)$$

The solution of Eq. (5.11) is both simple and efficient. Defining

$$\{\delta U^{**}\} \equiv [U]\{\delta U\} \quad (5.12)$$

a forward sweep is performed to solve

$$[L]\{\delta U^{**}\} = -\{R\} \quad (5.13)$$

for $\{\delta U^{**}\}$. This is followed by a backsweep to solve

$$[\mathcal{U}]\{\delta\mathbf{U}\} = \{\delta\mathbf{U}^{**}\} \quad (5.14)$$

for $\{\delta\mathbf{U}\}$. Since Eqs. (5.13) and (5.14) are triangular, the sweeps require only simple inversions of matrices.

A variant of the LU scheme, called the LU-SGS scheme, combines the advantages of the LU scheme with the robustness of a symmetric Gauss-Seidel (SGS) relaxation procedure. It was originally developed by Jameson and Yoon (1988) for the Euler and Navier-Stokes equations. It has since been applied to a variety of compressible and incompressible flow problems (Yoon, Kwak, and Chang, 1989; Yoon and Kwak, 1991).

The LU-SGS scheme can be derived by first writing the system of equations as

$$[\mathcal{D}]\{\delta\mathbf{U}\} + [\mathcal{L}]\{\delta\mathbf{U}\} + [\mathcal{U}]\{\delta\mathbf{U}\} = -\{\mathbf{R}\} \quad (5.15)$$

where $[\mathcal{M}]$ has been split into three operators

$$[\mathcal{M}] = [\mathcal{D}] + [\mathcal{L}] + [\mathcal{U}] \quad (5.16)$$

Here, $[\mathcal{D}]$ contains the diagonal terms, and $[\mathcal{L}]$ and $[\mathcal{U}]$ contain the off-diagonal lower and upper triangular terms, respectively. Eq. (5.15) can be solved using the two-sweep, SGS relaxation scheme,

$$[\mathcal{D}]\{\delta\mathbf{U}^{p+\frac{1}{2}}\} + [\mathcal{L}]\{\delta\mathbf{U}^{p+\frac{1}{2}}\} + [\mathcal{U}]\{\delta\mathbf{U}^p\} = -\{\mathbf{R}\} \quad (5.17)$$

$$[\mathcal{D}]\{\delta\mathbf{U}^{p+1}\} + [\mathcal{L}]\{\delta\mathbf{U}^{p+\frac{1}{2}}\} + [\mathcal{U}]\{\delta\mathbf{U}^{p+1}\} = -\{\mathbf{R}\} \quad (5.18)$$

where the superscript p is the iteration index. By rearranging terms, the above can be written as

$$([\mathcal{D}] + [\mathcal{L}]) \{\delta \mathbf{U}^{p+\frac{1}{2}}\} = -\{\mathbf{R}\} - [\mathcal{U}] \{\delta \mathbf{U}^p\} \quad (5.19)$$

$$([\mathcal{D}] + [\mathcal{U}]) \{\delta \mathbf{U}^{p+1}\} = -\{\mathbf{R}\} - [\mathcal{U}] \{\delta \mathbf{U}^{p+\frac{1}{2}}\} \quad (5.20)$$

Letting $p = 0$ and $\delta \mathbf{U}^0 = 0$, then from Eq. (5.19)

$$[\mathcal{D}] \{\delta \mathbf{U}^{p+\frac{1}{2}}\} = -\{\mathbf{R}\} - [\mathcal{U}] \{\delta \mathbf{U}^p\} \quad (5.21)$$

Substituting this relation into Eq. (5.20) yields

$$([\mathcal{D}] + [\mathcal{L}]) \{\delta \mathbf{U}^{p+\frac{1}{2}}\} = -\{\mathbf{R}\} \quad (5.22)$$

$$([\mathcal{D}] + [\mathcal{U}]) \{\delta \mathbf{U}^{p+1}\} = [\mathcal{D}] \{\delta \mathbf{U}^{p+\frac{1}{2}}\} \quad (5.23)$$

If only a single iteration is taken, then the above becomes the LU-SGS, which can be written as

$$([\mathcal{D}] + [\mathcal{L}]) \{\delta \mathbf{U}^{**}\} = -\{\mathbf{R}\} \quad (5.24)$$

$$([\mathcal{D}] + [\mathcal{U}]) \{\delta \mathbf{U}\} = [\mathcal{D}] \{\delta \mathbf{U}^{**}\} \quad (5.25)$$

From the preceding derivation, it can be seen that the LU-SGS scheme is equivalent to a single iteration of a SGS relaxation scheme with the solution vector $\{\delta \mathbf{U}\}$ initialized to zero.

For equations of the form given by Eq. (5.10), the $[\mathcal{D}]$, $[\mathcal{L}]$, and $[\mathcal{U}]$ operators can be defined in terms of the following point matrices:

$$[\mathcal{D}] = \frac{\Delta\tau}{\Omega} [\mathbf{I}_\theta] [\mathbf{C}_P] \quad (5.26)$$

$$[\mathcal{L}] = \frac{\Delta\tau}{\Omega} [\mathbf{I}_\theta] ([\mathbf{C}_W] \mathcal{E}_{i-1} + [\mathbf{C}_S] \mathcal{E}_{j-1} + [\mathbf{C}_B] \mathcal{E}_{k-1}) \quad (5.27)$$

$$[\mathcal{U}] = \frac{\Delta\tau}{\Omega} [\mathbf{I}_\theta] ([\mathbf{C}_E] \mathcal{E}_{i+1} + [\mathbf{C}_N] \mathcal{E}_{j+1} + [\mathbf{C}_F] \mathcal{E}_{k+1}) \quad (5.28)$$

Here, $\mathcal{E}_{i\pm 1}$, $\mathcal{E}_{j\pm 1}$, and $\mathcal{E}_{k\pm 1}$ denote shift operators (e.g. $\mathcal{E}_{i+1}\delta\mathbf{U} = \delta\mathbf{U}_{i+1}$). Using these matrices, the LU-SGS algorithm can be implemented as follows:

1. Forward Sweep

```

do k = 2 to kmax - 1
  do j = 2 to jmax - 1
    do i = 2 to imax - 1
       $\delta\mathbf{U}^{**} = -[\mathcal{D}]^{-1}(\{\mathbf{R}\} + [\mathcal{L}]\delta\mathbf{U}^{**})$ 
    end do
  end do
end do

```

2. Backward Sweep

```

do k = kmax - 1 to 2

```

```

do j = jmax - 1 to 2
  do i = imax - 1 to 2
     $\delta\mathbf{U} = \delta\mathbf{U}^{**} - [\mathbf{D}]^{-1}[\mathbf{U}]\delta\mathbf{U}$ 
  end do
end do
end do

```

5.5 The LGS Scheme

The LGS scheme solves Eq. (5.1) by forming a block tridiagonal system of equations along a particular coordinate line and solving this system for a tentative solution. This operation is repeated for all coordinate lines using successive forward or backward sweeps for each coordinate direction. The solution of block tridiagonal equations is carried out in the present work using the periodic and non-periodic FORTRAN subroutines provided in Anderson, Tannehill, and Pletcher (1984).

For a given coordinate line, the coefficients for the tridiagonal system are formed by retaining only those coefficients which lie along the coordinate line and placing all remaining terms on the right hand side. The right hand side terms are evaluated using the latest values of $\delta\mathbf{U}$. As an example, consider solving Eq. (5.9) along the ξ coordinate direction. The coefficients of the tridiagonal system can be written as

$$[A]\delta\mathbf{U}_{i-1} + [B]\delta\mathbf{U}_i + [C]\delta\mathbf{U}_{i+1} = D \quad (5.29)$$

where

$$[A] = [C_W] \quad (5.30)$$

$$[B] = [I] + [H] + [C_P] \quad (5.31)$$

$$[C] = [C_E] \quad (5.32)$$

$$D = R' - ([C_N]\delta U_{j+1} + [C_S]\delta U_{j-1} + [C_F]\delta U_{k+1} + [C_B]\delta U_{k-1}) \quad (5.33)$$

In this case, the coefficients are defined for $2 < i < imax - 1$. Equations for the other coordinate directions are analogous.

For completeness, a full specification of LGS algorithm is given below. Note that this algorithm assumes a single periodic boundary in the ξ direction at $i = 1$ and $i = imax$. In addition, only a single forward sweep is taken in each direction, although multiple forward/backward sweeps can be implemented if desired.

1. ξ Sweep

```

do k = 2 to kmax - 1
  do j = 2 to jmax - 1
    do i = 2 to imax - 1
      Construct coefficients [A], [B], [C], D
    end do
    Solve periodic tridiagonal system for  $\delta U$ .
  end do
end do

```

```

end do
end do

```

2. η Sweep

```

do i = 2 to imax - 1
  do k = 2 to kmax - 1
    do j = 2 to jmax - 1
      Construct coefficients [A], [B], [C], D
    end do
    Solve non-periodic tridiagonal system for  $\delta U$ .
  end do
end do

```

3. ζ Sweep

```

do i = 2 to imax - 1
  do j = 2 to jmax - 1
    do k = 2 to kmax - 1
      Construct coefficients [A], [B], [C], D
    end do
    Solve non-periodic tridiagonal system for  $\delta U$ .
  end do
end do

```

6. RESULTS

6.1 Introduction

In order to assess the accuracy, efficiency, and robustness of the surface capturing approach, extensive calculations were carried out for several two and three-dimensional validation problems. Results for the following four validation problems are presented in this chapter:

- The two-dimensional broken dam problem.
- The two-dimensional Rayleigh-Taylor instability problem.
- The three-dimensional broken dam problem.
- Axisymmetric spin-up in a spherical tank.

Separate computer programs were written to compute the two-dimensional and three-dimensional problems. These programs were given the names TFL2D and TFL3D, respectively. The faster turn around time for TFL2D reduced the effort involved in debugging various aspects of the methodology, and allowed extensive parametric studies to be carried out on the available computing facilities. The knowledge gained in the two-dimensional investigations was later employed in TFL3D.

Although the general single step, trapezoidal formulation presented in Chapter 4 was coded in both flow solvers, all calculations were performed using the Crank-Nicolson scheme ($\theta = \frac{1}{2}$) in order to provide second order accuracy in time. In addition, exploratory calculations revealed that while the use of pseudo-time terms in the mass and momentum equations did promote the stability of the scheme for a given physical time step, it also tended to slow down the convergence of the sub-iteration process. Consequently, all calculations were performed without the pseudo-time terms in the mass and momentum equations. The faster convergence afforded by this arrangement appeared to compensate for the smaller physical time step required for stability.

Both TFL2D and TFL3D were written in the FORTRAN 77 programming language, and were designed to be compiled and run on UNIX-based workstations. All calculations were performed on an HP 9000/730 workstation with 32 MB of RAM running the HP-UX 8.07 operating system. While a few calculations were also made on the NASA NAS Cray supercomputers, it was found that the turnaround times for the problems considered were not significantly better than those for the workstation. Moreover, the workstation environment allowed much more control to be exercised over the compilation and execution of the computer programs. The performance of the computer programs on supercomputers could have been greatly enhanced by modifying specific subroutines to take advantage of vectorization. Although such optimizations were not pursued in the present work, they would clearly be desirable for production-level three-dimensional calculations, which would likely consume large amounts of computer CPU time and storage.

Fig. 6.1 illustrates the basic input/output design that was employed in the TFL2D and TFL3D programs. As shown in this figure, the grid and initial flowfield files required by the flow solvers were created by two auxiliary computer programs. Both grid point and flowfield data files were saved in a format compatible with the PLOT3D CFD postprocessing program (Walatka et al., 1990). The flow solver also required a user-supplied input file which contained input and output file names, reference data, and solution control parameters. The output from the flow solvers included both PLOT3D-compatible flowfield solution files and other data files containing solution-derived data (e.g. velocity, pressure, or density profiles). The flowfield solution files could also be used to restart the flow solver, if necessary.

In interpreting the solutions obtained using the surface capturing approach, it was important to unambiguously define the location of the free surface based on the density distribution within the computational domain. Unlike surface fitting methods, the location of the free surface can only be approximated within the resolution provided by the grid. In the present work, the position of the free surface was defined as the surface of constant density for which $\rho(x, y, z) = \frac{1}{2}(\rho_1 + \rho_2) \equiv \rho_m$. All free surface plots presented in the sections below were created using ten evenly-spaced contour levels between $0.9\rho_m$ and $1.1\rho_m$.

6.2 The Two-dimensional Broken Dam Problem

The sudden collapse of a rectangular column of fluid onto a horizontal surface is a classic problem in free surface hydrodynamics (Stoker, 1958). Due to its use in modeling the sudden failure of a dam, it has become known as the broken dam problem. The relatively simple geometry and initial conditions associated with the

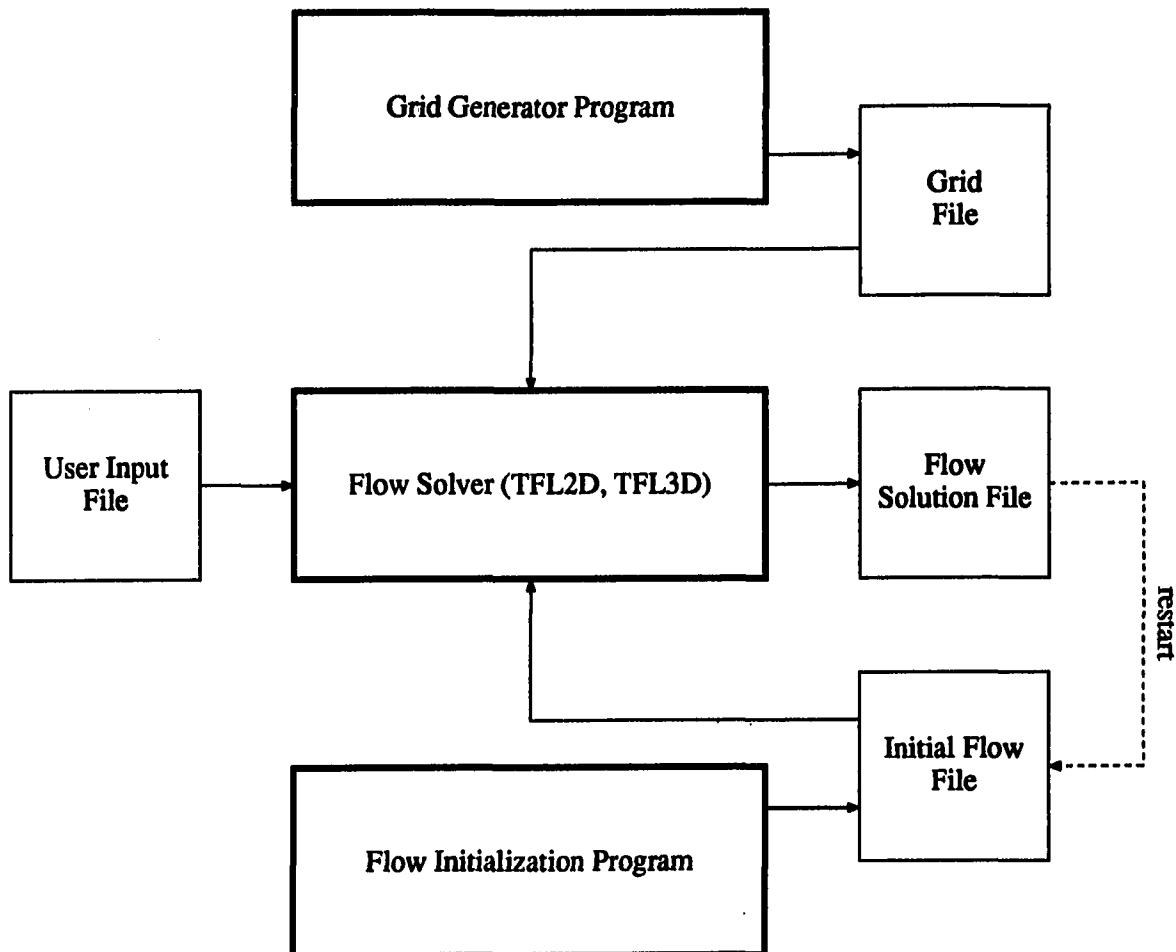


Figure 6.1: Schematic of computer program design

broken dam problem has made it a popular validation case for various surface tracking and surface capturing schemes. Some examples of such calculations can be found in Hirt and Nichols (1981), Maxwell and Spalding (1987), and Jun and Spalding (1987).

The specific geometry employed in the present work is illustrated in Fig. 6.2. A square water column a units wide is enclosed within an air-filled container $5a$ units long by $1.25a$ units high. The water is initially retained by a thin partition (the dam) on the right hand side of the column. At time $t = 0+$, the partition is removed, thereby allowing the water to collapse under the influence of gravity.

The specific configuration shown in Fig. 6.2 was chosen in order to approximate the collapsing water column experiments of Martin and Moyce (1952). For their experiments, Martin and Moyce constructed a 0.05715 m wide Perspex channel with a $0.05715\text{ m} \times 0.05715\text{ m} \times 0.127\text{ m}$ fluid reservoir at one end. A water column was held initially in the reservoir by a thin waxed paper diaphragm affixed to a metal heater strip. The collapse of the water column was initiated by applying a large electrical current to the heater strip, which, in turn, caused the diaphragm to detach. A high speed camera was then used to photograph the subsequent fluid motion at selected time intervals. From these photographs, Martin and Moyce were able to tabulate the surge front position x_s and the column height y_s as functions of time.

Aside from the use of a two-dimensional model, the present geometry differs from the Martin and Moyce experiments in that the container in Fig. 6.2 is closed, whereas the channel used in the Martin and Moyce experiments was open to the atmosphere. However, since the density of water is much larger than the density of air, the use of a closed container should not have a significant effect on the motion of the water.

The initial conditions for the calculations were prescribed as follows. The density

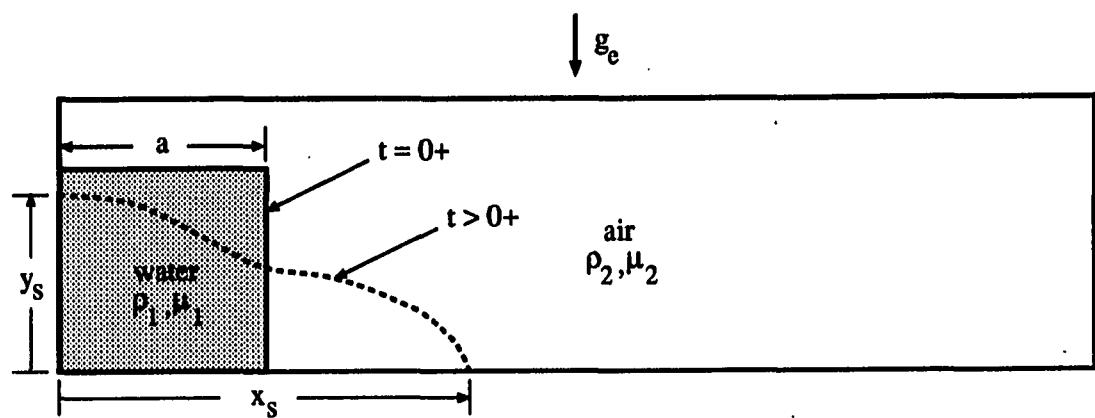


Figure 6.2: Illustration of the two-dimensional broken dam problem

field was initialized with values appropriate for each fluid as shown in Fig. 6.2, the velocity components were set to zero everywhere, and the pressure distribution was arbitrarily defined to be hydrostatic relative to the top surface of the liquid. It should be noted that since this problem involves an instantaneous change in the initial state of the two fluids (when the partition is removed), the true initial conditions at time $t = 0+$ will be different from those given above. The reason for this is that the hydrostatic pressure distributions along the each side of the partition must equilibrate at time $t = 0+$, which, in turn, induces a non-zero velocity field. This condition appears to develop naturally at the first time step of the numerical algorithm, and the calculations to date have not been adversely affected by the use of the simpler initial conditions.

A total of ten cases were computed for the present study. The key parameters associated with these cases are summarized in Table 6.1. Reference quantities for nondimensionalization were selected as follows: $L_r = a$, $V_r = \sqrt{g_e a}$, $\rho_r = \rho_1$, and $\mu_r = \mu_1$.

For the case of a square column of water, Martin and Moyce obtained data at two different length scales: $a = 0.05715\text{ m}$ and $a = 0.1143\text{ m}$. With the reference quantities defined above, the corresponding Reynolds numbers were 42792 and 121033. Both cases were examined in this study.

After extensive exploratory calculations, a baseline case (case 1 in Table 6.1) was established using a uniform grid of 80×20 control volumes and a length scale of $a = 0.05715\text{ m}$. These baseline results were then used as a reference for other cases. All calculations were allowed to proceed until the surge front reached the forward wall of the container (at a nondimensional time of approximately $t^* = 3$).

Table 6.1: Case summary for the two-dimensional broken dam problem

ρ_1 (kg/m ³)	ρ_2 (kg/m ³)	μ_1 (N - s/m ²)	μ_2 (N - s/m ²)	g_e (m/s ²)	β (N/m ²)	Fr	We
1000	1.21	1×10^{-3}	1.81×10^{-5}	9.81	5×10^4	1	∞

Case	Grid	Δt^*	Interpolant	Limiter (ρ)	Re
1	80×20	0.002	QUICK	comp. minmod	42792
2	120×30	0.00133	QUICK	comp. minmod	42792
3	160×40	0.001	QUICK	comp. minmod	42792
4	80×20	0.001	QUICK	comp. minmod	42792
5	80×20	0.002	QUICK	comp. minmod	121033
6	80×20	0.002	1 st order upwind	N/A	42792
7	80×20	0.002	2 nd order upwind	comp. minmod	42792
8	80×20	0.002	QUICK	minmod	42792
9	80×20	0.002	QUICK	van Leer	42792
10	80×20	0.002	QUICK	superbee	42792

To establish the grid independence of the solutions relative to the baseline case, calculations were made using uniform grids of 120×30 and 160×40 control volumes (cases 2 and 3). The time step was reduced in both cases to maintain a constant reference CFL number ($V_r \Delta t / \Delta x$). A separate case was also computed to demonstrate the effect of reducing the time step on the baseline grid (case 4).

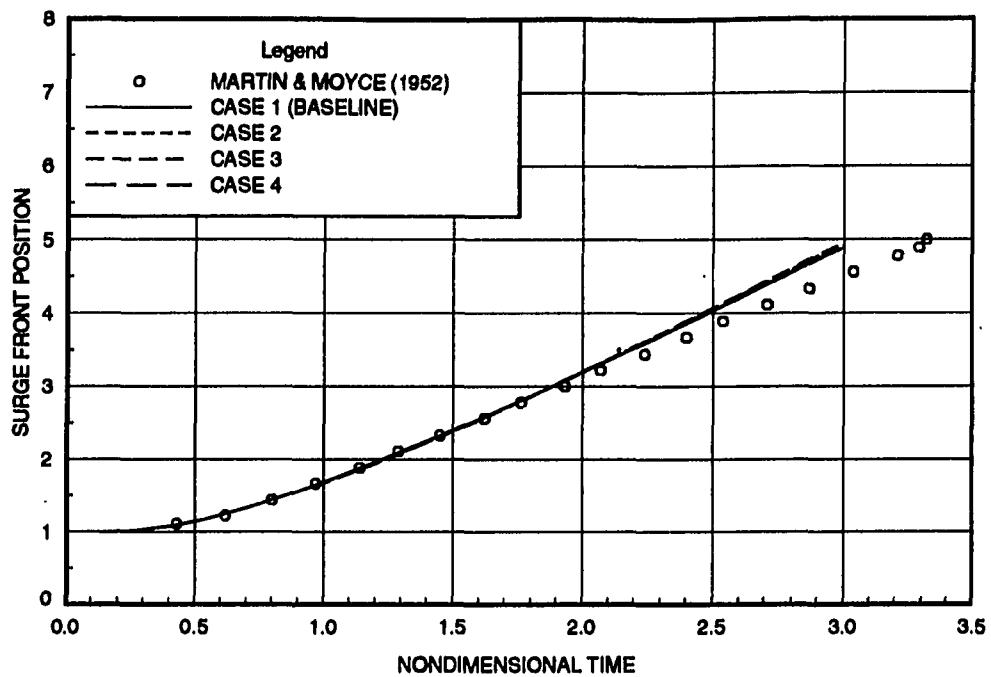
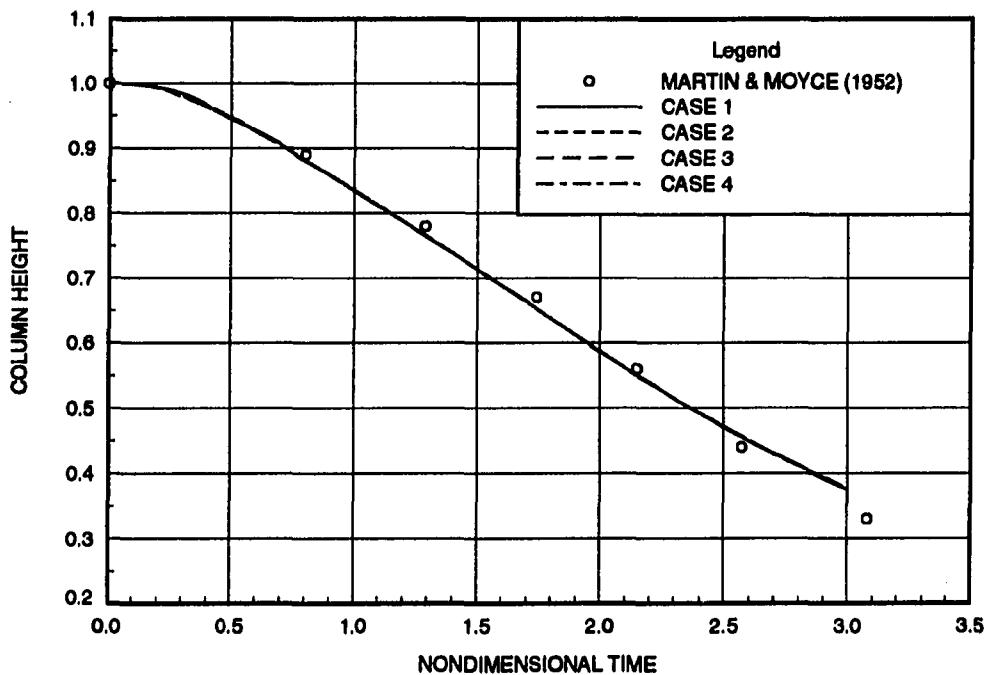
The computed surge front and column height positions (nondimensionalized with respect to a) for cases 1 - 4 are plotted with respect to the nondimensional time in Figs. 6.3 and 6.4. The data of Martin and Moyce (1952) are also plotted for comparison. From these plots, it can be seen that the baseline results are essentially grid independent. In addition, the numerical solutions compare well with experimental data, considering the uncertainties inherent in the experimental data and the approximations employed in the numerical model. However, the computed surge front position does appear to diverge somewhat from the experimental data towards the

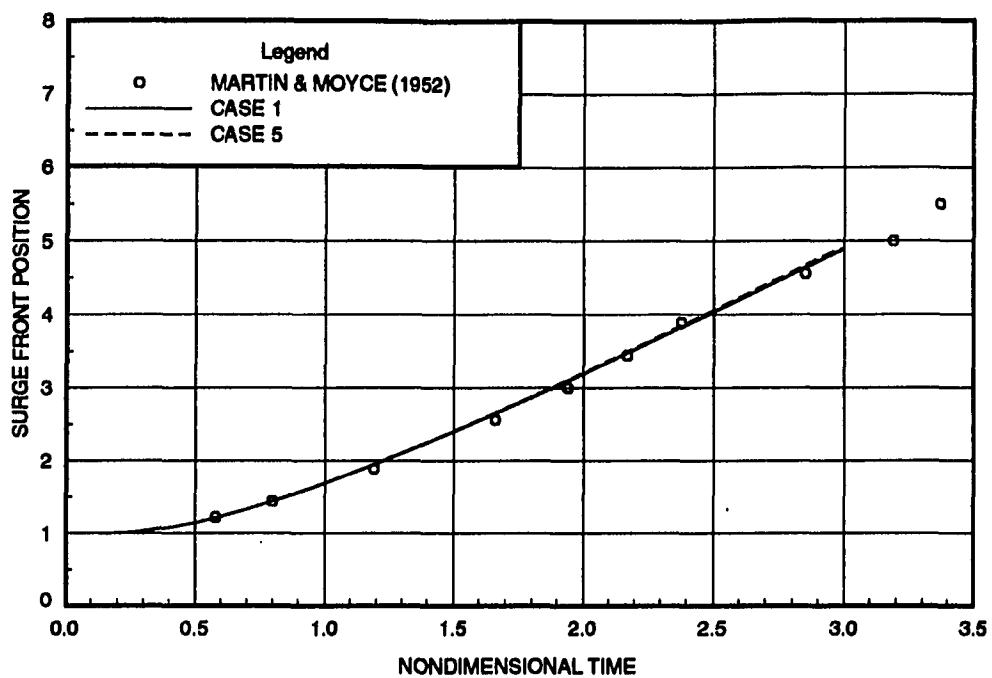
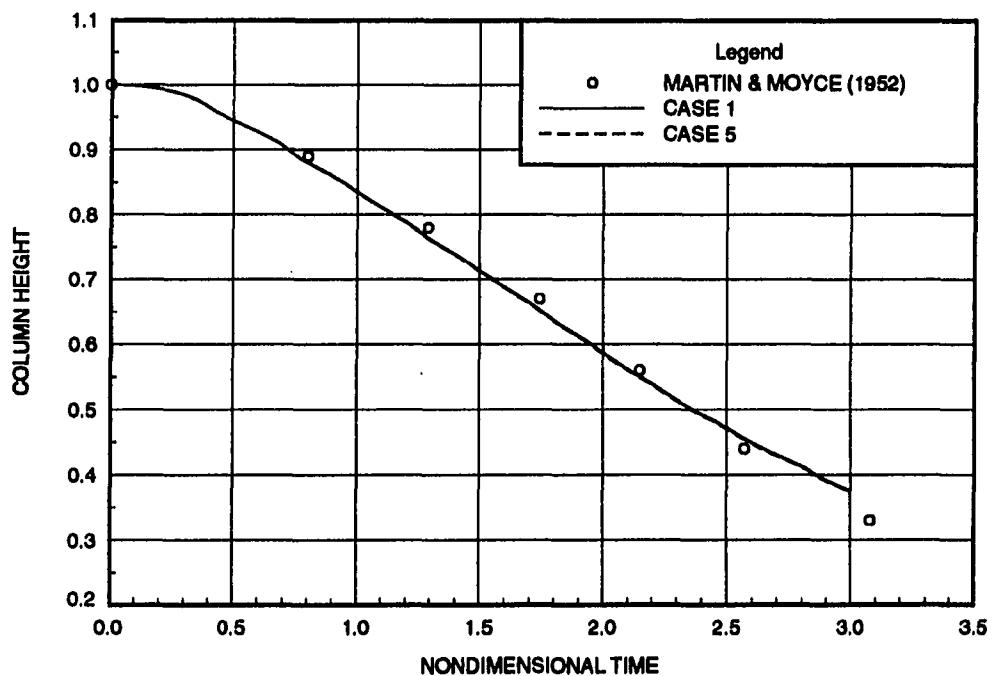
end of the transient.

To examine the effect of Reynolds number, a separate high Reynolds number calculation was performed on the baseline grid using $a = 0.1143\text{ m}$ (case 5). A comparison of these results with both the baseline case and the $a = 0.1143\text{ m}$ data of Martin and Moyce (1952) is shown in Figs. 6.5 and 6.6 (note that the Martin and Moyce data in Fig. 6.6 are for $a = 0.05715\text{ m}$ since measurements for this Reynolds number were not reported). The close agreement of the two numerical solutions suggests that increasing the Reynolds number relative to the baseline case does not significantly affect the free surface motion. Moreover, the high Reynold number results appear to agree more closely with the data of Martin and Moyce. It is not clear whether or not this indicates that the primary source of the discrepancy observed at the lower Reynolds number is due to uncertainties in the experimental data, or, perhaps, to some other effect not accounted for by the two-dimensional model.

The free surface and velocity fields for the 160×40 control volume grid (case 3) are plotted at selected times in Figs. 6.7 and 6.8. Initially, the large difference between the hydrostatic pressure distribution in the water relative to the adjacent air creates a large horizontal pressure gradient at the bottom of the dam. This, in turn, accelerates the fluid along the bottom of the container. Eventually, an elongated fluid layer develops as the surge front sweeps toward the front of the container. An interesting feature of this flowfield is the vortex which forms in the vicinity of the free surface. This vortex appears to be induced by the motion of the water relative to the air.

For comparison, a series of photographs from the Martin and Moyce experi-

Figure 6.3: Surge front position versus time for $a = 0.05715 \text{ m}$ Figure 6.4: Column height versus time for $a = 0.05715 \text{ m}$

Figure 6.5: Surge front position versus time for $a = 0.1143 \text{ m}$ Figure 6.6: Column height versus time for $a = 0.1143 \text{ m}$

ments are reproduced in Fig. 6.9. It can be seen that the free surface profiles in these photographs are quite similar to those predicted numerically. Unfortunately, a direct comparison of the free surface shapes could not be made because the times corresponding to each photograph were not indicated.

The effect of various upwind-biased interpolation schemes and the density limiter functions on the numerical solution were examined in cases 6 – 10. Comparisons of solution quality were made using the sharpness of the density profile at the water-air interface as a figure of merit. Experience with the surface capturing method has shown that it is important to prevent excessive diffusion of the density interface in order to maintain the accuracy of the solution for long time periods.

Since the velocity field is continuous for viscous flows, no limiters were applied to the interpolation of the velocity components. While pressure field itself is also continuous, there will be a jump in the pressure gradient across the free surface. To prevent any undesirable behavior, a simple minmod limiter was used in the interpolation of the pressure.

In Fig. 6.10, the density profiles along the bottom wall of the container at time $t^* = 1.5$ are plotted for solutions using first order upwind, second order upwind, and QUICK interpolants for all variables (see Section 3.4.4.2). Both the second order upwind and QUICK cases used a compressive minmod limiter for density (no limiting is necessary for the first order upwind interpolant). As expected, first order upwind interpolation produced a highly smeared density profile, whereas the higher order interpolants yielded substantially sharper profiles. Overall, the QUICK profile was the best of the three.

The impact of the limiter function on the density profile is shown in Fig. 6.11.

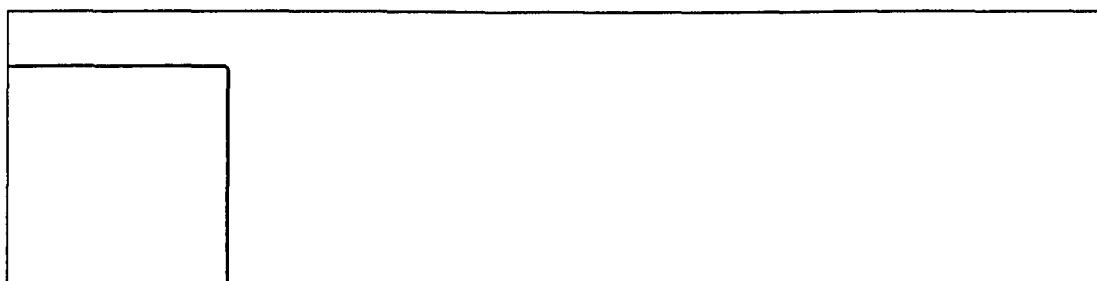
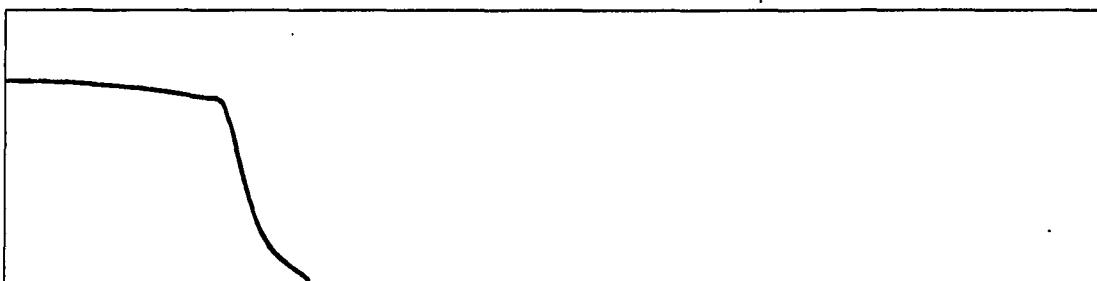
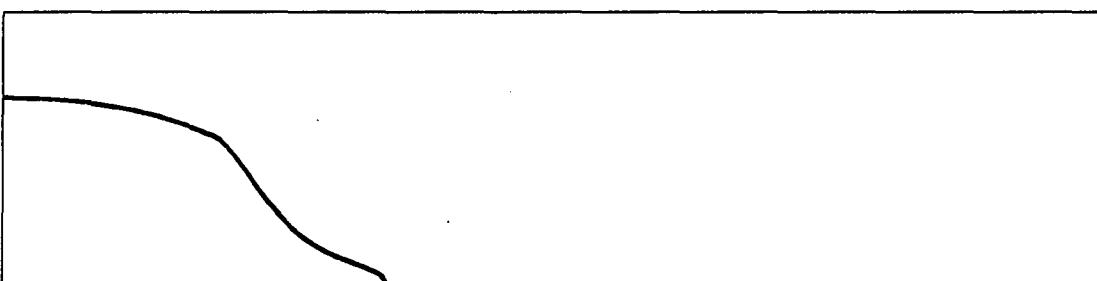
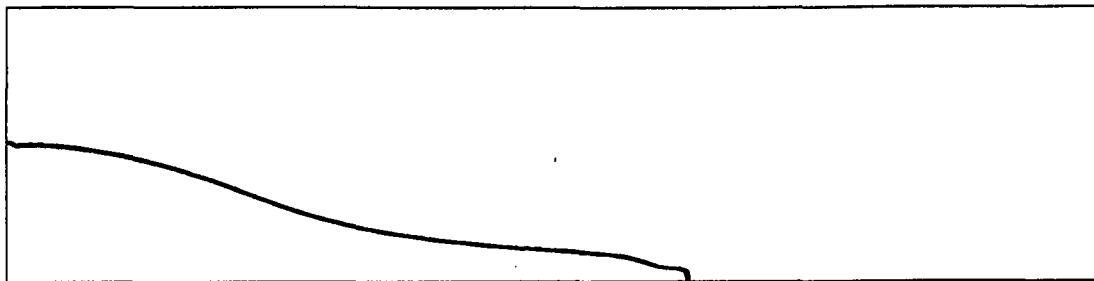
TIME = 0.0**TIME = 0.6****TIME = 1.2**

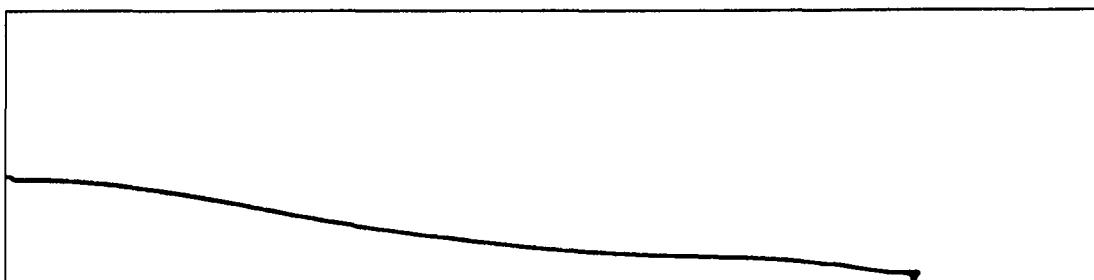
Figure 6.7: Free surface motion for two-dimensional broken dam problem

120

TIME = 1.8



TIME = 2.4



TIME = 3.0

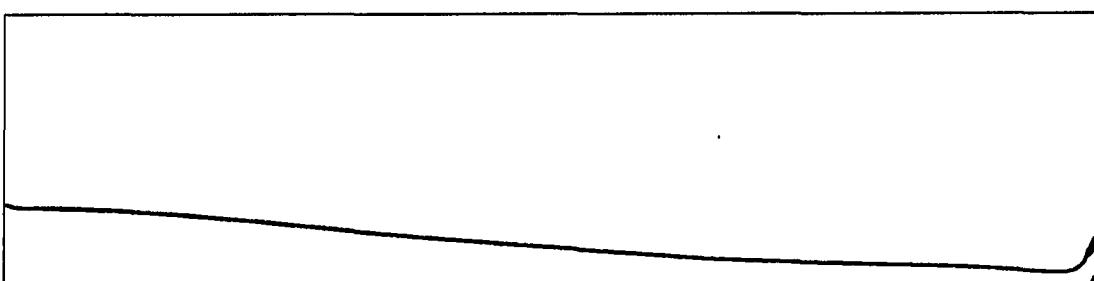


Fig. 6.7: (Continued)

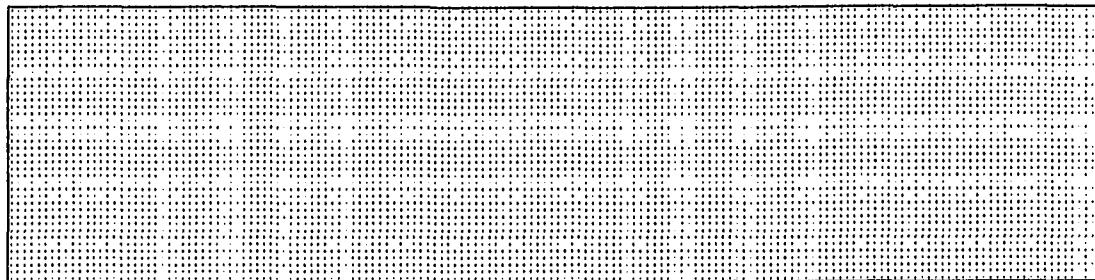
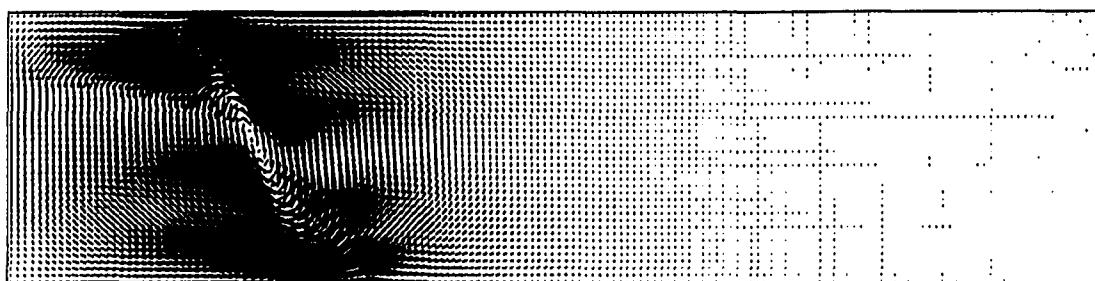
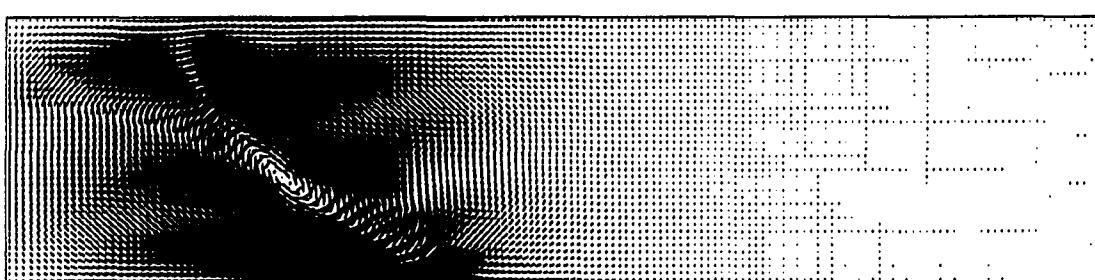
TIME = 0.0**TIME = 0.6****TIME = 1.2**

Figure 6.8: Velocity field for two-dimensional broken dam problem

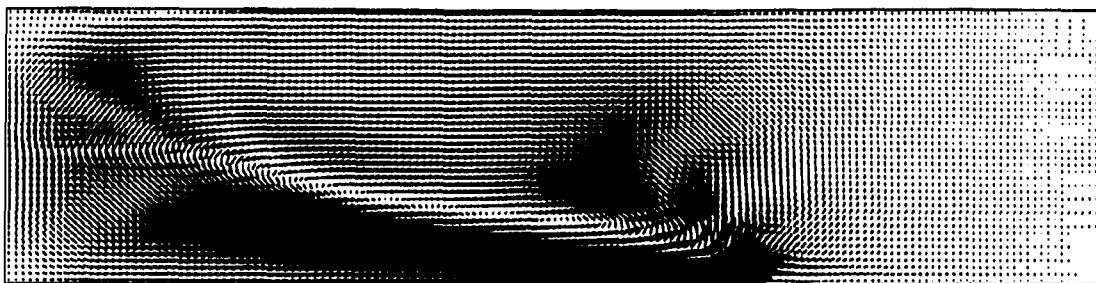
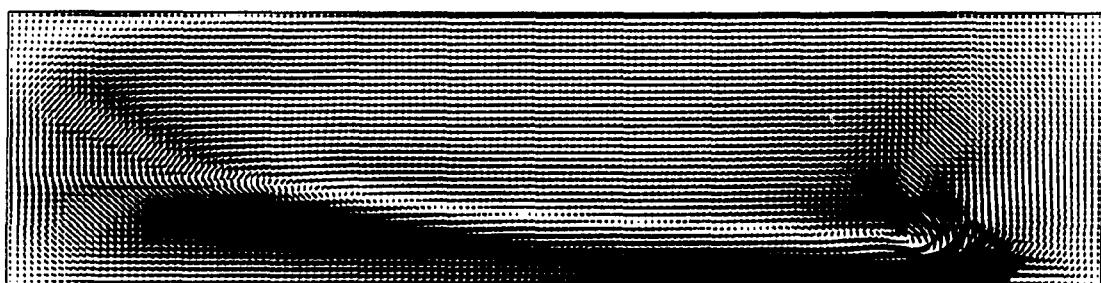
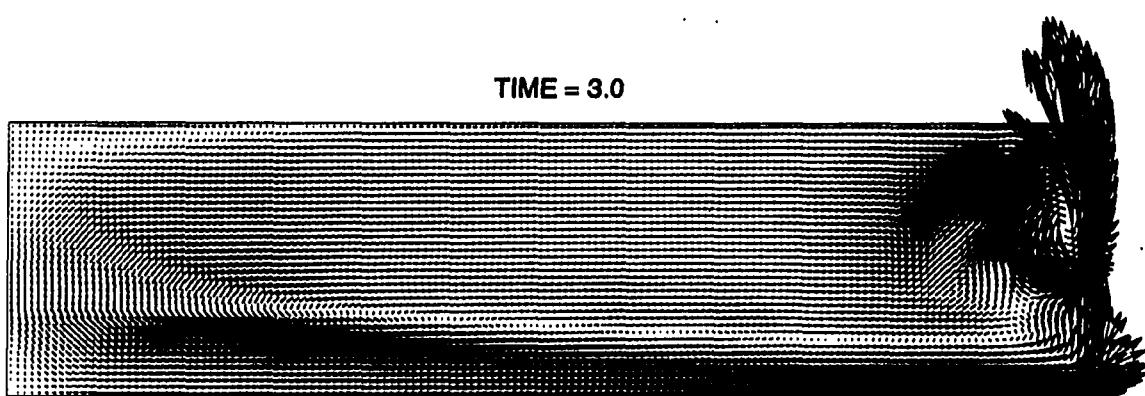
TIME = 1.8**TIME = 2.4****TIME = 3.0**

Fig. 6.8: (Continued)

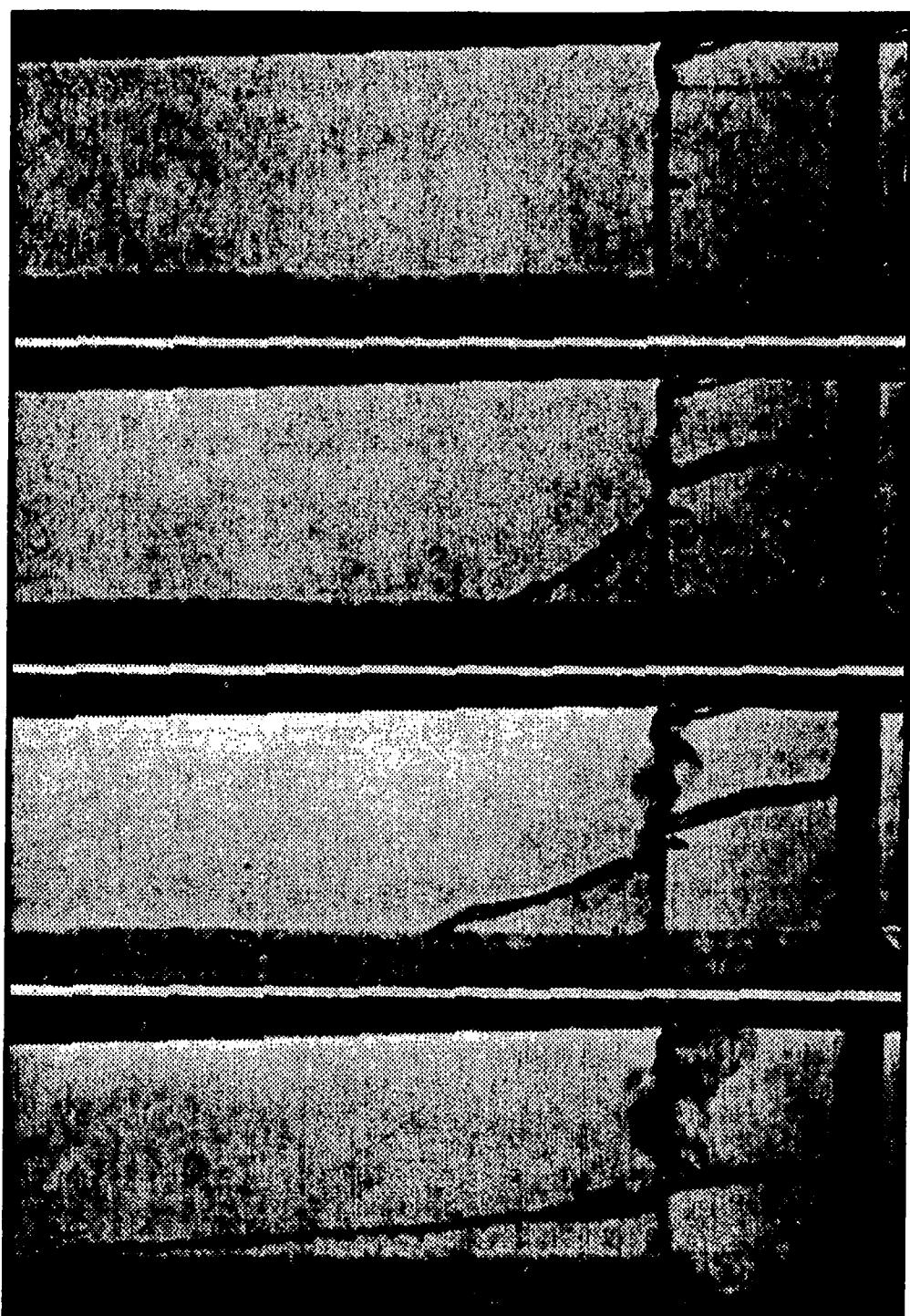


Figure 6.9: Photographs of broken dam experiment (from Martin and Moyce, 1952)

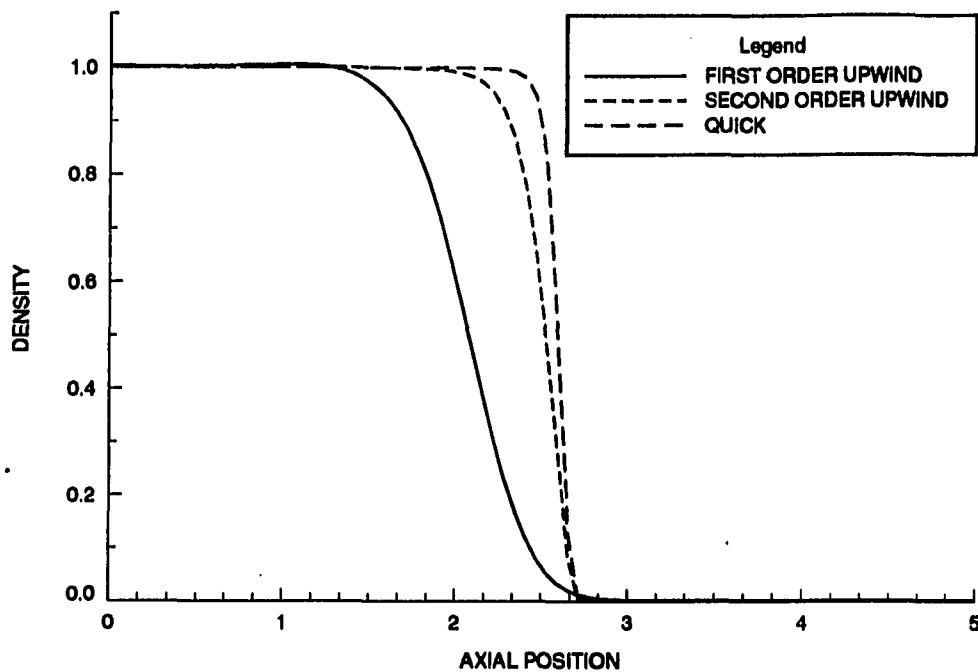


Figure 6.10: Comparison of density profiles for first and third order interpolants

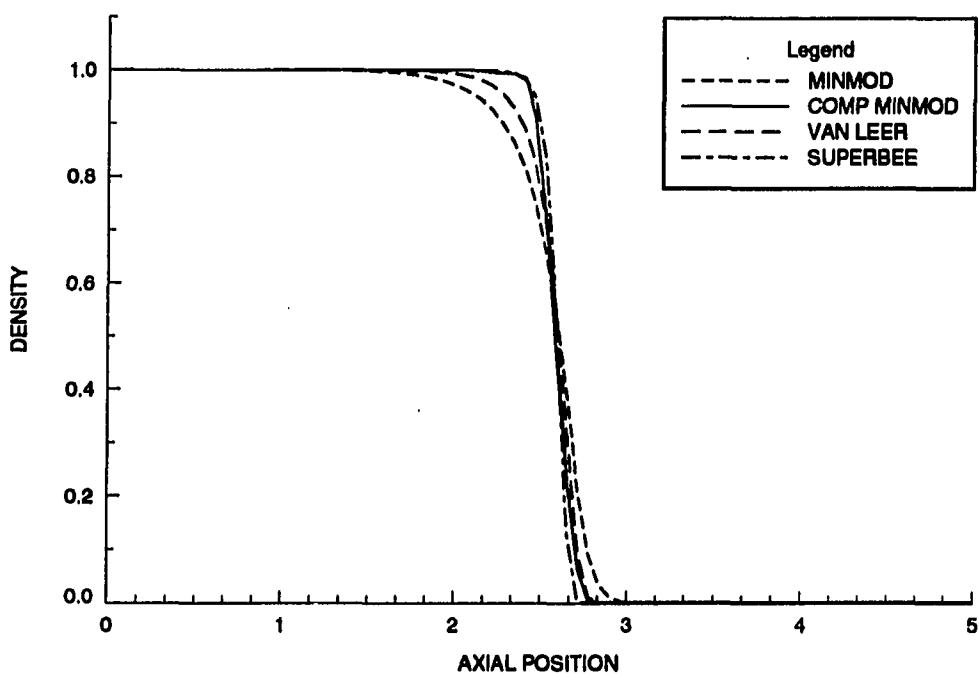


Figure 6.11: Comparison of density profiles for various limiters

Again, the density profiles along the bottom wall of the container at time $t^* = 1.5$ are plotted. It can be seen that the minmod limiter is the most diffusive and the superbee limiter the least. While these results would seem to favor the use of the superbee limiter, experience has shown that calculations using the superbee limiter are not as robust as those for the other limiters. This behavior is consistent with results reported in the literature for explicit and implicit TVD MUSCL schemes (Yee, 1989). Fortunately, the compressive minmod limiter produced profiles which were comparable to the superbee limiter without the attendant robustness problems.

Based on the foregoing results, all subsequent two and three-dimensional calculations were performed using the QUICK interpolants for all variables, and the compressive minmod limiter for density. While this combination has proven adequate for the calculations performed to date, additional study would be required to determine if other combinations of interpolants and limiters (including some not listed here) would be superior to these.

For all of the broken dam cases, the conservation of total mass was monitored by summing the fluid mass ($\rho\Omega$) within each control volume over the entire computational domain. In every case, the change in total mass over the duration of the transient was less than 0.01%. Similar levels of mass conservation were maintained in all subsequent two and three-dimensional calculations.

Finally, an important link between the nondimensional pseudo-compressibility parameter β^* and the computed density field was revealed in the broken dam calculations. Specifically, it was found that if β^* was set to one, as is common in most incompressible calculations using the artificial compressibility method (see, for example, Rogers, Kwak, and Kiris, 1989), large, unphysical oscillations in the density

were created near the free surface. An analysis of a simple one-dimensional analog of the two-fluid equations confirmed this connection. Numerical experiments showed that this problem could be effectively suppressed by prescribing a larger value of β^* (generally on the order of $10^1 - 10^2$). However, making β^* too large tended to result in divergence of the pseudo-time marching process. It was concluded that an "optimal" value of β^* can be prescribed which provides sufficient suppression of density oscillations without significantly affecting the convergence properties of the numerical algorithm. To date, an appropriate value of β^* for a particular problem has been determined through trial and error.

6.3 The Two-dimensional Rayleigh-Taylor Instability

When a horizontal layer of heavy fluid overlies a layer of light fluid in the presence of a vertical gravitational field, the interface between the two fluids is unstable. If the interface is perturbed, buoyancy forces will cause the amplitude of the perturbation to grow with time. This phenomenon is known as the Rayleigh-Taylor instability.

The original problem, as formulated by Rayleigh (1900) and Taylor (1950), was concerned with a linear perturbation analysis for the case of two inviscid, incompressible fluid layers. The validity of the linear solutions at early times in the transient was later confirmed by the experiments of Lewis (1950). Other investigators extended the linear analysis to include the effects of viscosity and surface tension (Chandrasekhar, 1961). More recently, CFD methods have been used to obtain numerical solutions to the full, nonlinear problem. Some noteworthy examples of such calculations can be found in the works of Daly (1967), Hirt, Cook, and Butler (1970), Youngs (1984), Tryggvason (1988), and Mulder, Osher, and Sethian (1992).

The present work considers the Rayleigh-Taylor instability for two viscous, incompressible fluid layers with a prescribed density ratio (ρ_1/ρ_2) of two and a uniform kinematic viscosity ($\nu_1 = \nu_2$). As illustrated in Fig. 6.12, the fluids are confined within a periodic domain of width $2L$ and height H which is bounded above and below by impermeable walls. The flowfield is assumed to be symmetric about $x = 0$.

A single wavelength perturbation is introduced at the fluid interface using the following nondimensional initial velocity field, which was adapted from the work of Daly (1967):

$$u^* = \begin{cases} \alpha \sin(\pi x^*) \exp(-\pi|y^*|) & y^* > 0 \\ -\alpha \sin(\pi x^*) \exp(-\pi|y^*|) & y^* < 0 \end{cases} \quad (6.1)$$

$$v^* = \alpha \cos(\pi x^*) \exp(-\pi|y^*|) \quad (6.2)$$

$$\alpha = \frac{\pi A \Delta y^*}{2V_r} \quad (6.3)$$

Here, A is a perturbation amplitude, and Δy^* is a representative mesh increment in the vertical direction. This velocity field corresponds to a sinusoidal perturbation of wavelength $2L$. To complete the specification of the initial conditions, the density field was prescribed as shown in Fig. 6.12, while the initial pressure was set to a hydrostatic distribution.

Four cases were run initially to obtain solutions for Reynolds numbers of 28.3 and 283 on uniform grids of 30×90 and 40×120 control volumes. The key parameters for these four cases are presented in Table 6.2. All calculations were carried out for 4800 time steps, with $\Delta t^* = 0.001$. Reference quantities for nondimensionalization

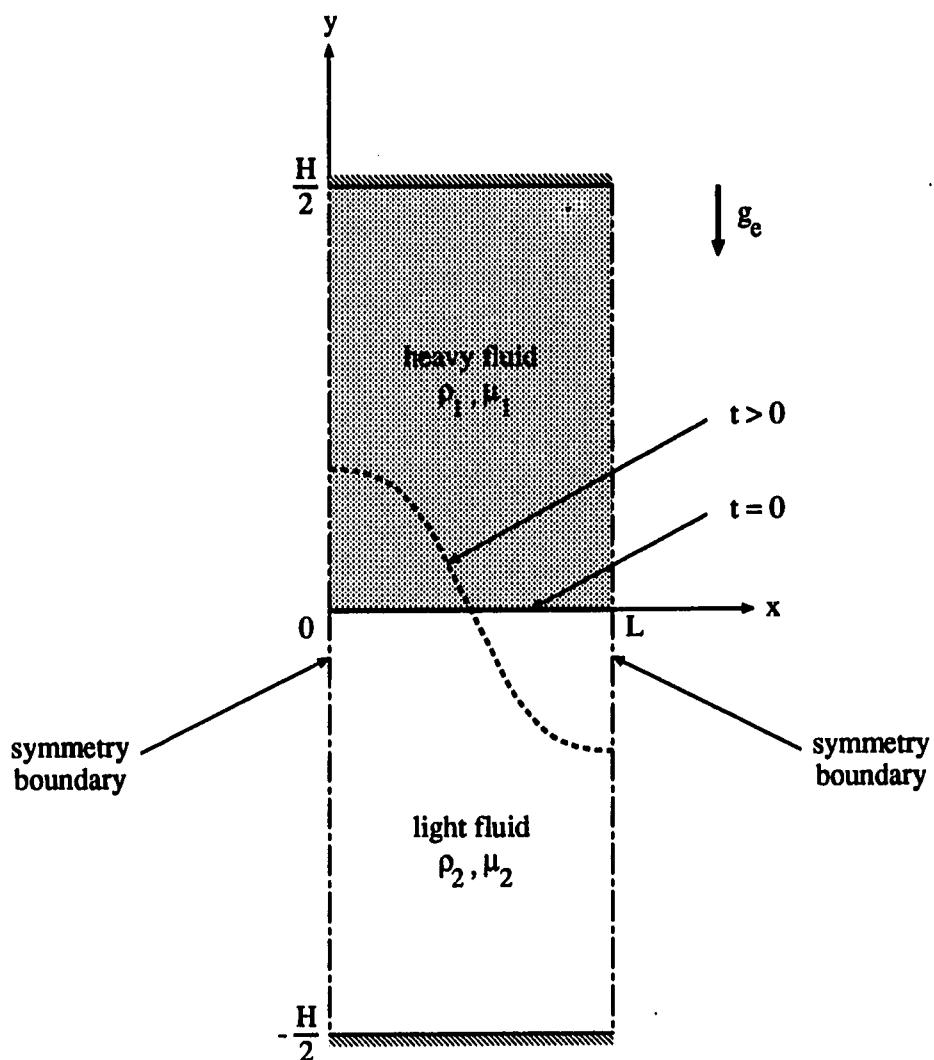


Figure 6.12: Illustration of the Rayleigh-Taylor instability problem

Table 6.2: Case summary for Rayleigh-Taylor instability: set #1

ρ_1 (kg/m ³)	ρ_2 (kg/m ³)	L (m)	H (m)	A (m/s)	g_e (m/s ²)	β (N/m ²)	Fr	We
2	1	0.02	0.06	1	1	1×10^3	1	∞

Case	Grid	μ_1 (N - s/m ²)	μ_2 (N - s/m ²)	Δt^*	Re
1	30×90	2×10^{-4}	1×10^{-4}	0.001	28.3
2	30×90	2×10^{-5}	1×10^{-5}	0.001	283
3	40×120	2×10^{-4}	1×10^{-4}	0.001	28.3
4	40×120	2×10^{-5}	1×10^{-5}	0.001	283

were selected as follows: $L_r = L$, $V_r = \sqrt{g_e L}$, $\rho_r = \rho_1$, and $\mu_r = \mu_1$.

The positions of free surface at the left and right boundaries are plotted versus nondimensional time in Figs. 6.13 and 6.13. It is seen that the speed of the interface motion along both boundaries is higher for the high Reynolds number case. This trend is expected since the larger viscous stresses associated with low Reynolds numbers would retard the acceleration of the fluid. The discrepancy between the curves for the two mesh sizes is due to the use of the mesh increment Δy^* in defining the initial velocity perturbation. Notice, however, that the behavior of the corresponding curves is similar.

The evolution of the free surface and the velocity field for the low and high Reynolds number cases is illustrated in Figs. 6.15 – 6.18 (results for the 40×120 control volume grid are shown). In both cases, the initial perturbation causes the light fluid to rise along the left boundary, while the heavy fluid sinks along the right boundary. The displacement of the interface is seen to be nearly symmetric during the early growth phase of the instability. As the amplitude of the instability increases, the characteristic mushroom shape emerges in the vicinity of the central vortex. The

rollup of the interface is much more pronounced for the higher Reynolds number case due to the smaller influence of viscous effects, which would tend to smooth out sharp velocity gradients. Eventually, the walls begin to influence the solution during the latter stages of the transient, especially in the high Reynolds number case. To permit the interface to develop unimpeded for longer times, a larger computational domain would be required.

According to linear theory, viscosity has a significant effect on the growth rate of a single wavelength perturbation. Specifically, it is found that as the perturbation wavelength decreases, the growth rate increases without bound in the absence of viscosity, while the growth rate for the viscous case approaches zero. At large perturbation wavelengths, however, the predicted growth rates for both the inviscid and viscous cases approach zero. It follows that a maximum growth rate must exist for the viscous case at some intermediate perturbation wavelength.

For the specific case of equal kinematic viscosities ($\nu_1 = \nu_2 = \nu$), Chandrasekhar (1961) derived a relation between the linear growth rate and the perturbation wavelength. This variation can be expressed in terms of a nondimensional growth rate $n^* = n\nu^{1/3}/g_e^{2/3}$ and a modified Reynolds number $Re_m = \lambda_p^{3/2}g_e^{1/2}/\nu$, where n is the dimensional growth rate (with units of $1/s$) and $\lambda_p = 2L$ is the perturbation wavelength. From this result, Chandrasekhar was able to determine the maximum growth rate and its corresponding wavelength for a given density ratio, kinematic viscosity, and gravitational acceleration. Additional details of Chandrasekhar's solution are provided in Appendix G.

To determine if the present surface capturing method could reproduce the growth rates predicted by linear analysis, a series of calculations was performed for a density

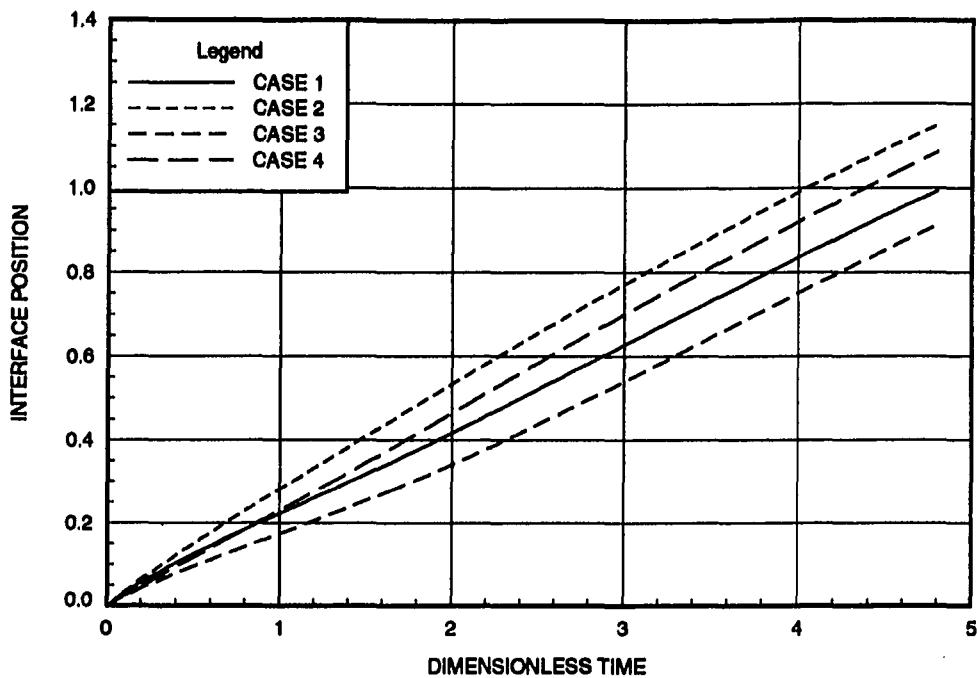


Figure 6.13: Time history of the free surface motion along the left boundary

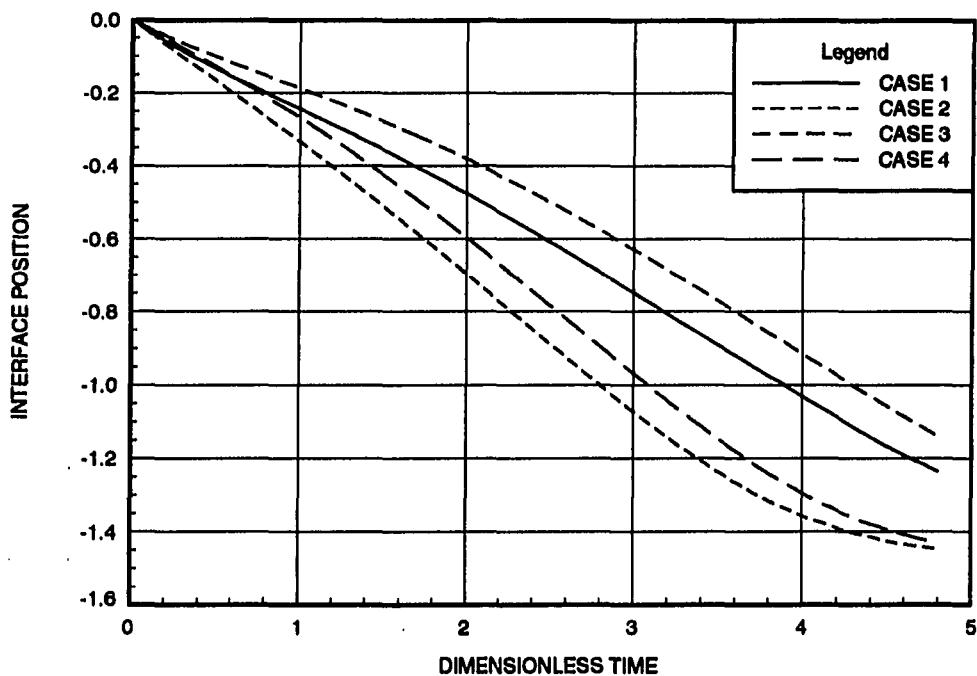


Figure 6.14: Time history of the free surface motion along the right boundary

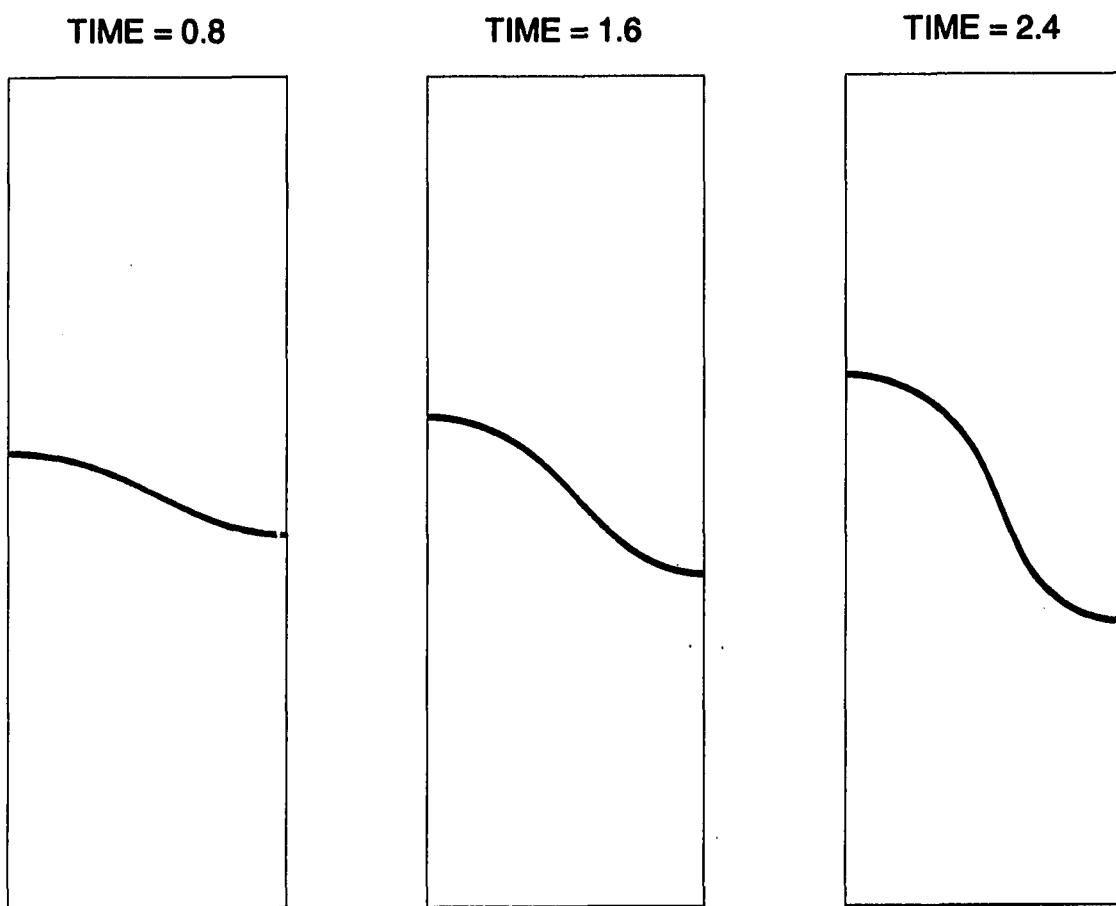


Figure 6.15: Free surface motion for the Rayleigh-Taylor instability : $Re = 28.3$

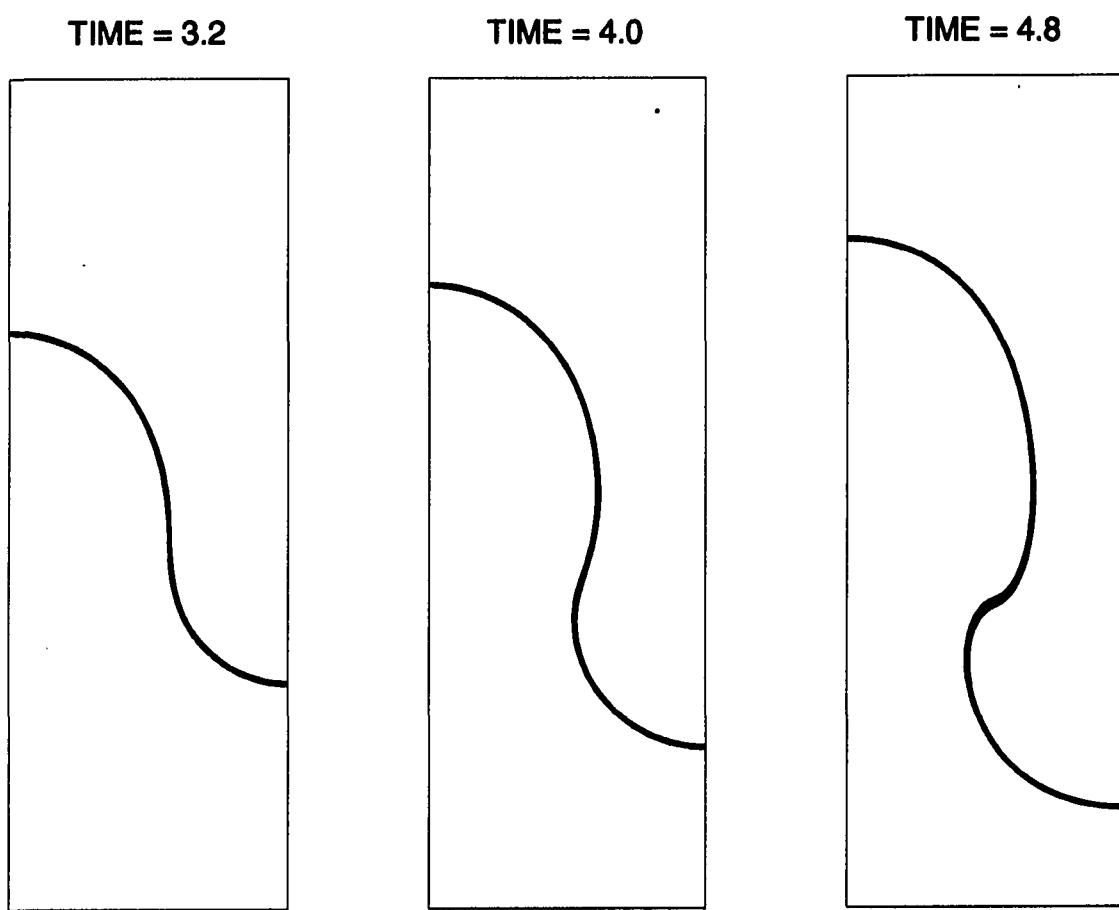


Fig. 6.15: (Continued)

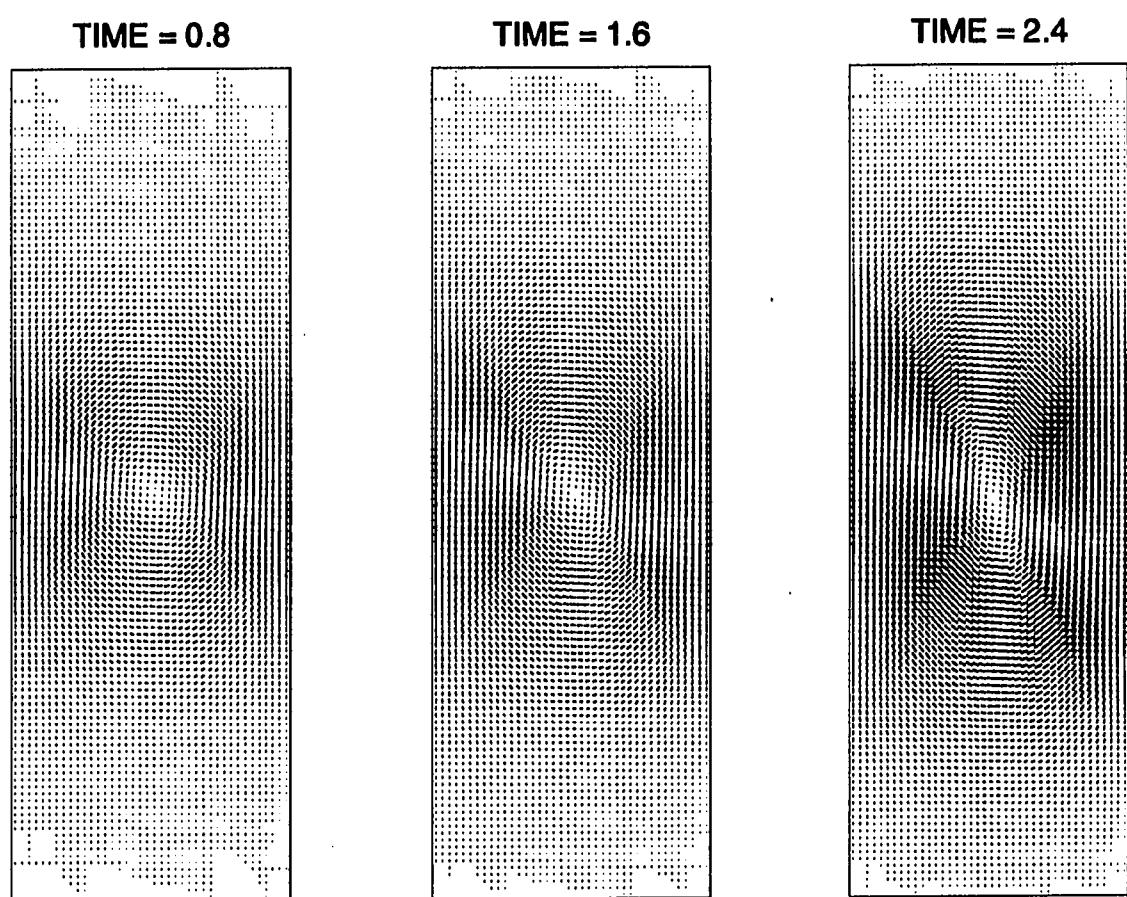


Figure 6.16: Velocity field for the Rayleigh-Taylor instability : $Re = 28.3$

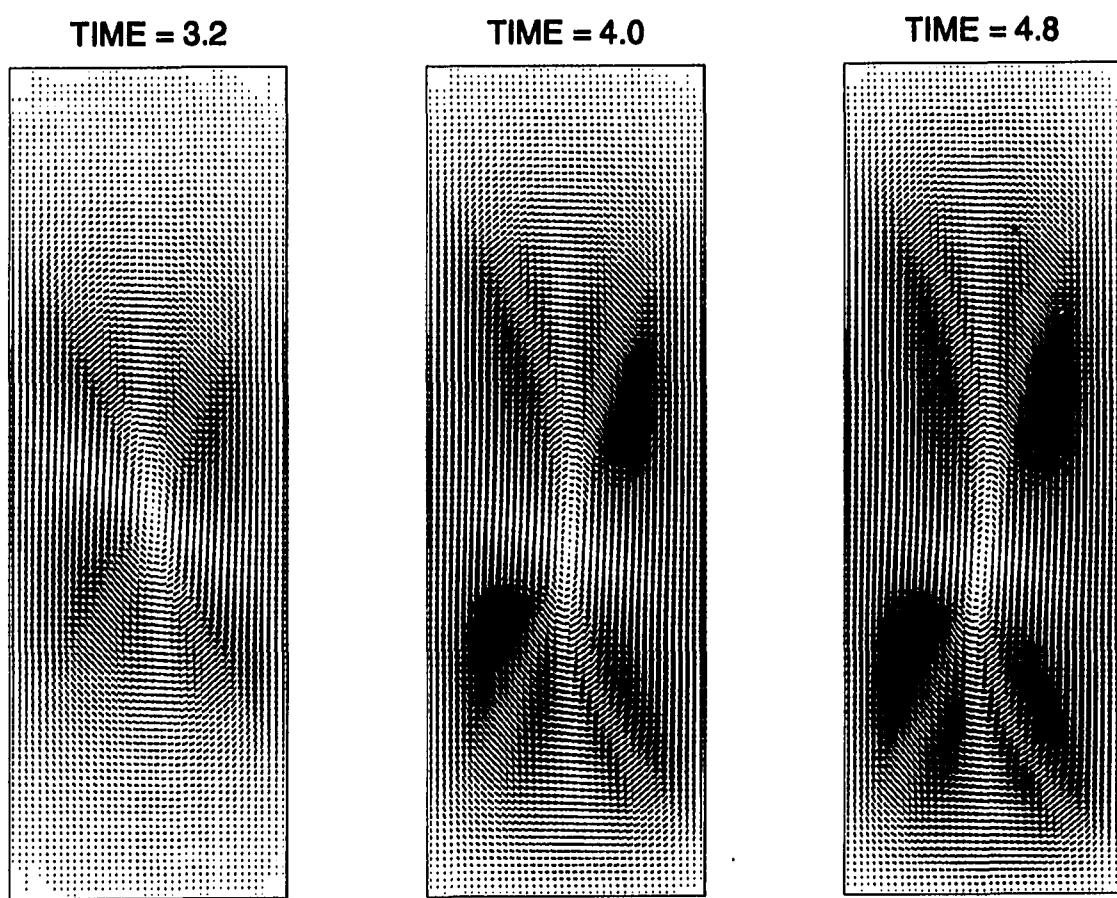


Fig. 6.16: (Continued)

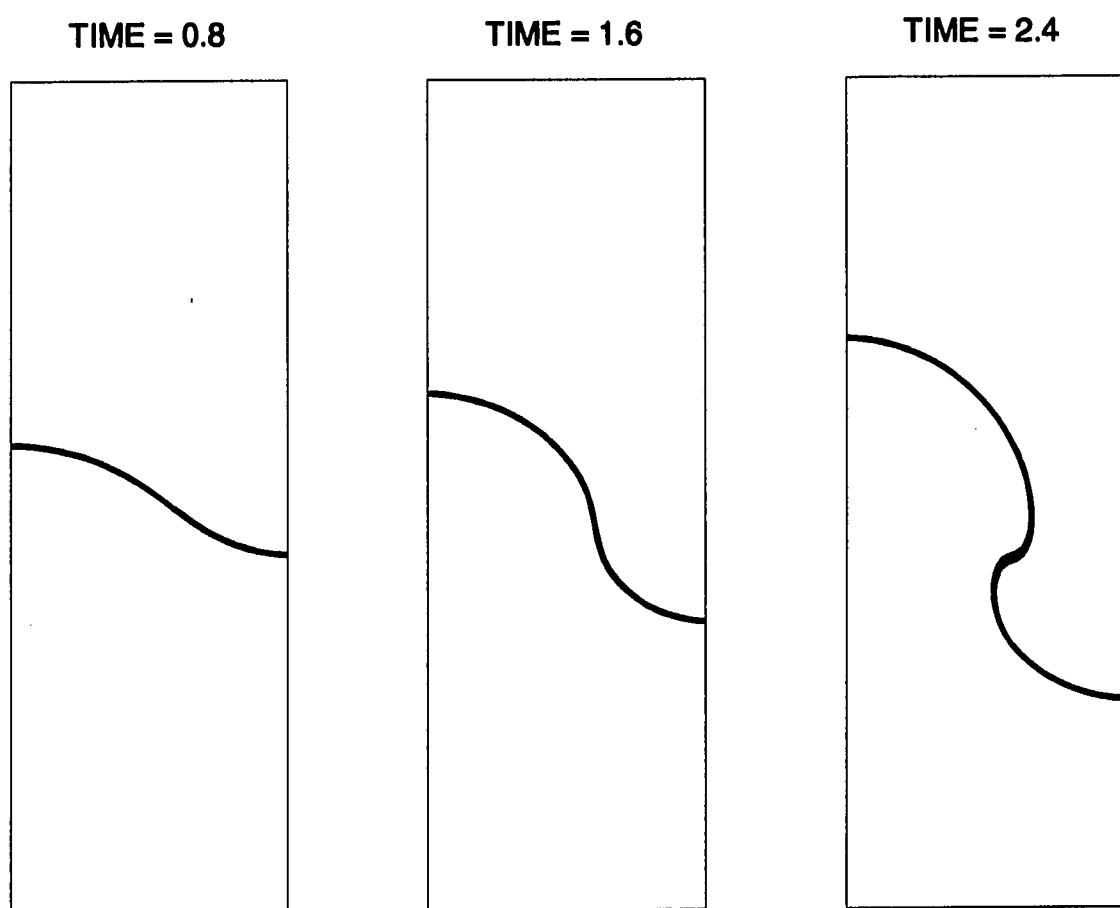


Figure 6.17: Free surface motion for the Rayleigh-Taylor instability : $Re = 283$

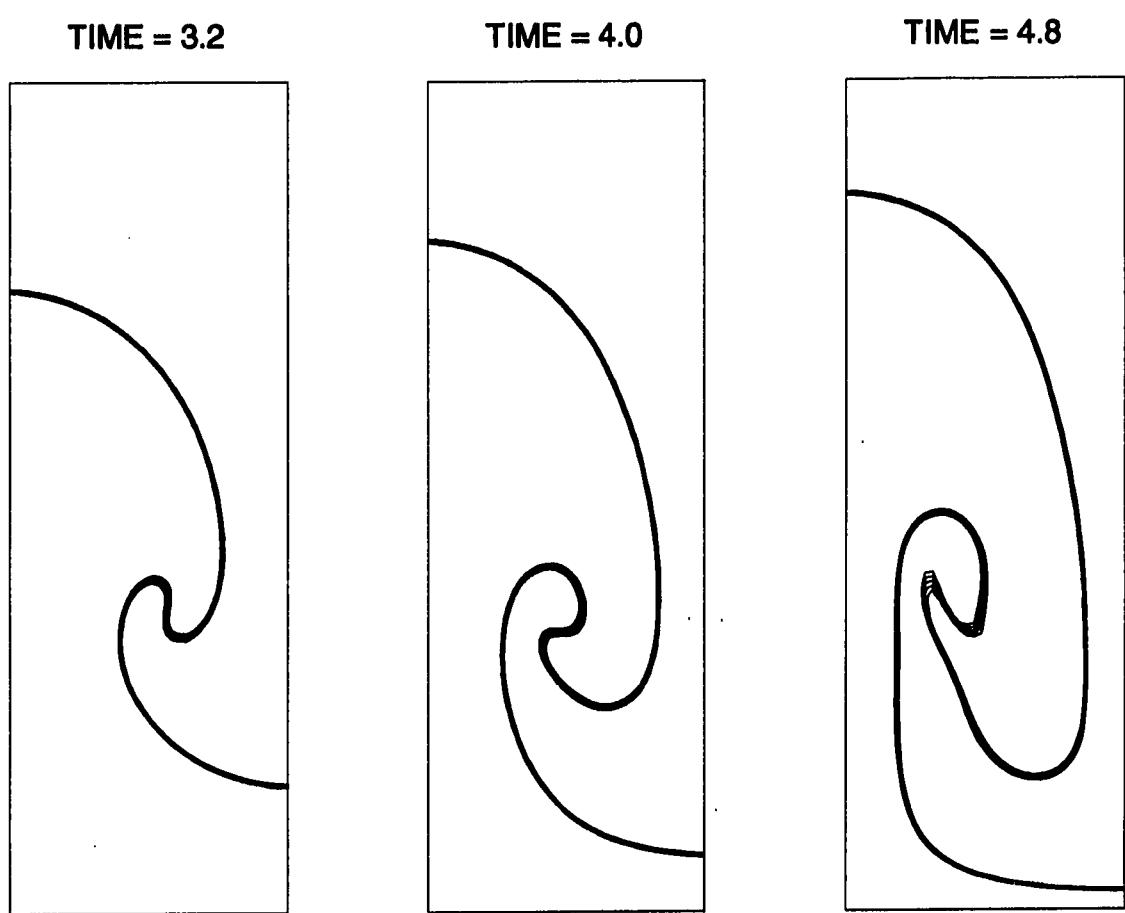


Fig. 6.17: (Continued)

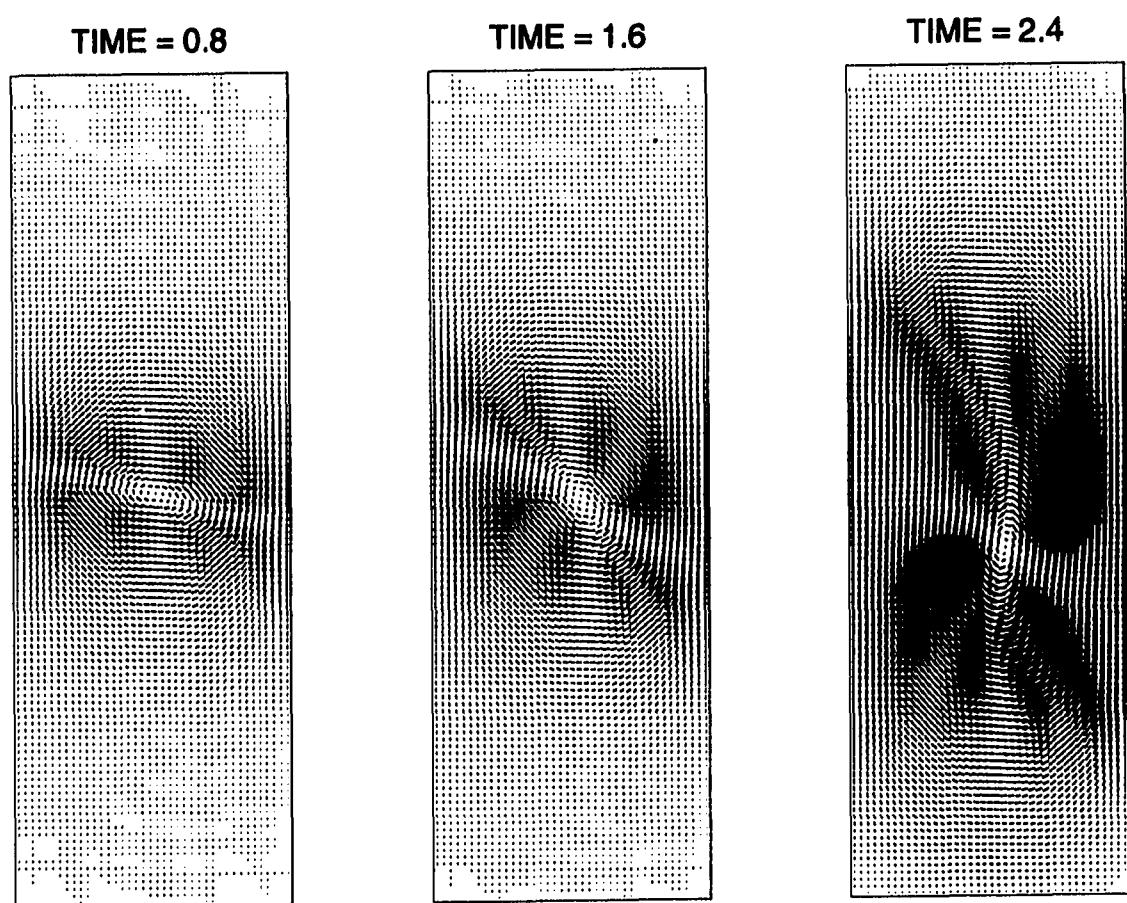


Figure 6.18: Velocity field for the Rayleigh-Taylor instability : $Re = 283$

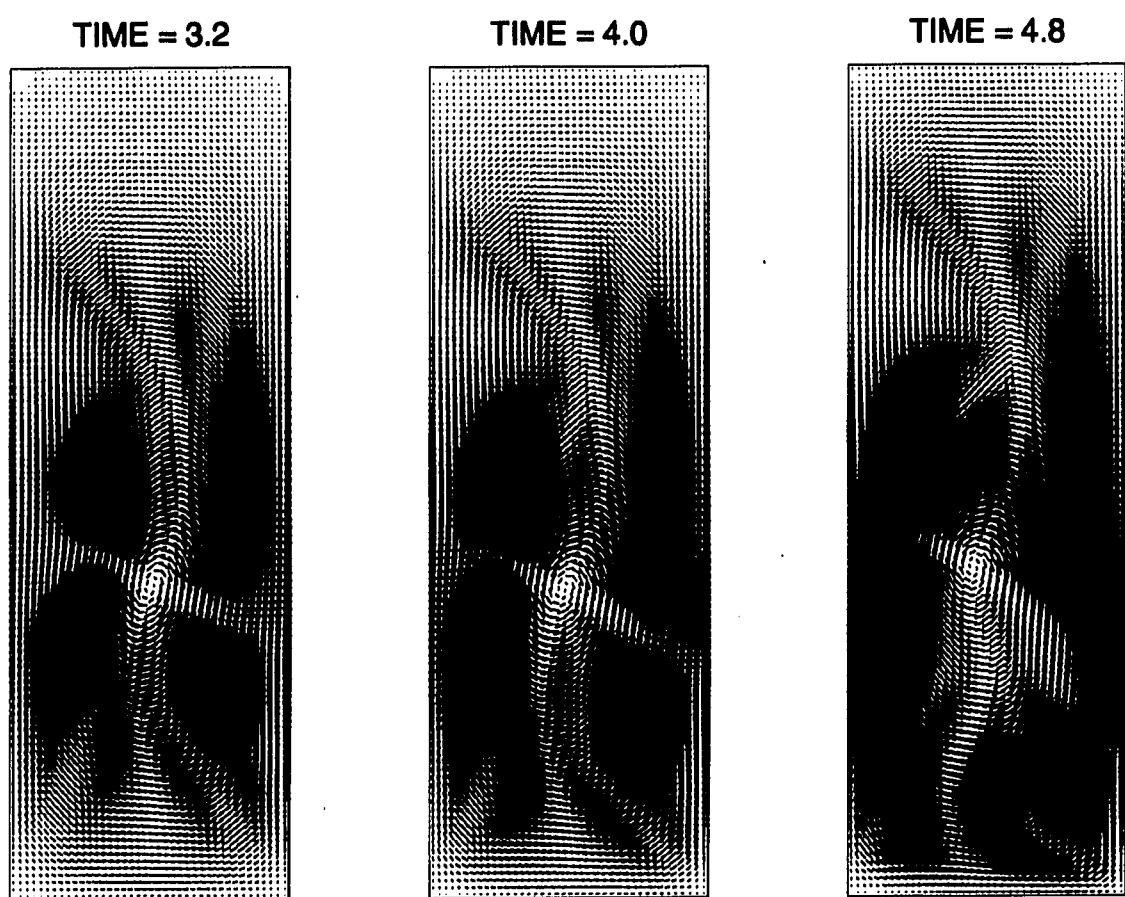


Fig. 6.18: (Continued)

ratio of two using four values of kinematic viscosity to provide a range of modified Reynolds numbers. The perturbation amplitude for these calculations was reduced by an order of magnitude ($A = 0.1$) to help maintain the linear behavior of the solution over the duration of the transient.

The growth rate for each case was calculated from the numerical results using a procedure adapted from the work of Daly (1967). This procedure is based on the fact that the amplitude of the interface displacement grows exponentially in time such that, after initial transients have died away, a single positive growth rate emerges. When this occurs, the natural logarithm of the interface displacement should vary linearly with time, the slope of this line being the growth rate. This leads to the following procedure for deriving the growth rate:

1. From a given numerical solution, compute and tabulate the left boundary interface position (y_L) and the right boundary interface position (y_R) as functions of time.
2. Plot both $\ln(|y_L|)$ and $\ln(|y_R|)$ versus time and deduce the growth rates n_L and n_R from the slopes of the plotted curves. The slopes should be measured at times following the initial transient.
3. Compute the perturbation growth rate by averaging n_L and n_R .

It should be noted that during the linear growth phase, the perturbation grows symmetrically, and thus n_L and n_R should be identical. However, since linear conditions can only be approximated in the numerical simulation, there will be a slight discrepancy between the computed growth rates n_L and n_R , especially at later times.

Table 6.3: Case summary for Rayleigh-Taylor instability: set #2

ρ_1 (kg/m ³)	ρ_2 (kg/m ³)	L (m)	H (m)	A (m/s)	g_e (m/s ²)	β (N/m ²)	Fr	We
2	1	0.02	0.06	0.1	1	1×10^3	1	∞

Case	Grid	μ_1 (N - s/m ²)	μ_2 (N - s/m ²)	Re_m
1	30×90	8×10^{-4}	4×10^{-4}	20
2	30×90	2×10^{-4}	1×10^{-4}	80
3	30×90	8×10^{-5}	4×10^{-5}	200
4	30×90	2×10^{-5}	1×10^{-5}	800

It is therefore appropriate to average the computed growth rates to obtain a single, representative growth rate for the interface.

The calculations for all four cases were performed on a uniform 30×90 control volume grid. The key parameters are summarized in Table 6.3.

The natural logarithm of the computed interface amplitudes at the left and right boundaries are plotted as functions of time in Fig. 6.19. Each curve is characterized by an initial transient followed by a positive exponential growth phase. The slope of linear region of each curve was obtained by performing a least squares fit on the data in the linear region. This value was then used to determine the growth rate as described above. A comparison of the computed growth rates with the growth rate variation predicted by the linear analysis of Chandrasekhar is given in Fig. 6.20. The numerical results show good agreement with the theoretical curve, and appear to reproduce the expected decrease in growth rate at small and large modified Reynolds numbers. The minor discrepancies exhibited by the numerical results can be attributed to errors associated with computing the growth rate and the slight departure of the numerical solution from linear behavior.

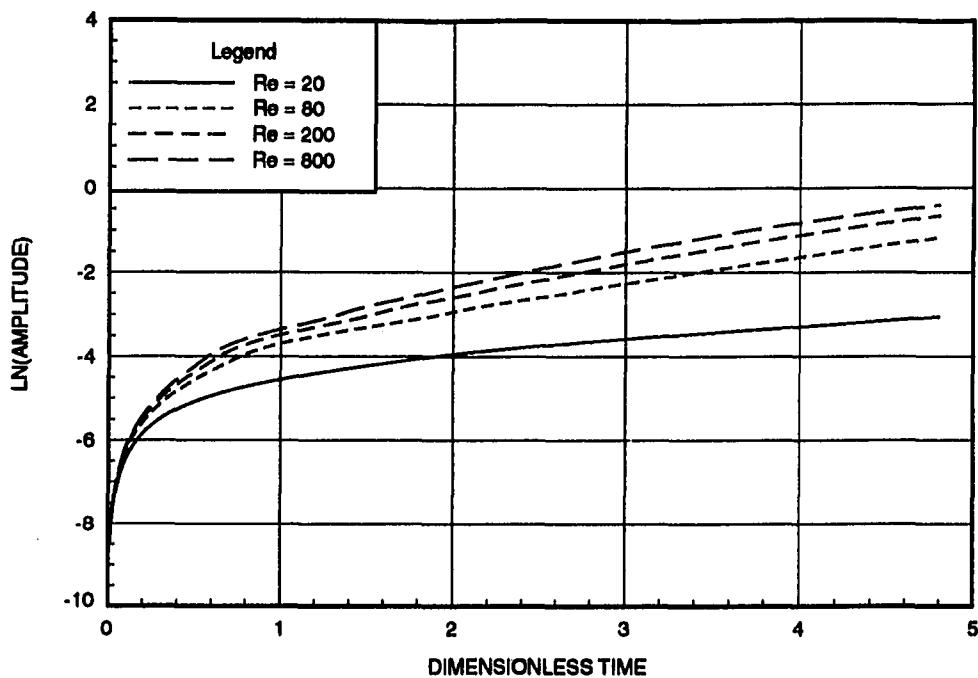


Figure 6.19: Evolution of average perturbation amplitude for several modified Reynolds numbers

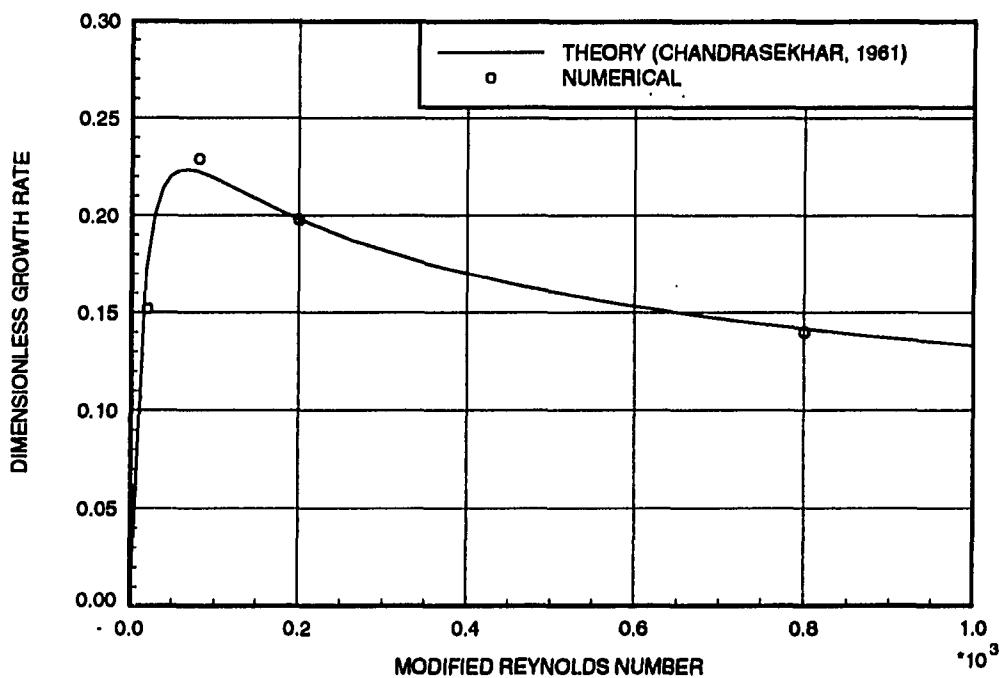


Figure 6.20: Nondimensional perturbation growth rate versus modified Reynolds number: comparison of numerical results with linear theory

6.4 The Three-dimensional Broken Dam Problem

As a first test case for the three-dimensional surface capturing algorithm, calculations were carried out for a three-dimensional version of the broken dam problem. As illustrated in Fig. 6.21, the geometry is essentially the same as in the two-dimensional case, except that the container is now assumed to have a finite width a in the z coordinate direction. Due to the symmetry of the flowfield, only half of the channel was used as the computational domain, with a symmetry plane established at $z = 0.5a$.

The calculations were made using a uniform grid of $80 \times 20 \times 10$ control volumes and a length scale of $a = 0.05715\text{ m}$. The initial conditions and reference quantities for nondimensionalization were identical to those employed in the two-dimensional case. Other key parameters for the calculation are summarized in Table 6.4.

Plots of the surge front and column height positions at the symmetry plane of the channel are presented in Figs. 6.22 and 6.23. Also plotted in these figures are the experimental data of Martin and Moyce (1952) and the corresponding results for the two-dimensional broken dam problem. From these plots, it can be seen that there is very little difference between the two and three-dimensional solutions. This is probably due to the high Reynolds number for this particular case, which reduced the influence of the side walls on the flowfield at the symmetry plane.

The motion of the free surface within the half channel is depicted in Fig. 6.24. Here, density contours lines were plotted on each boundary surface to provide an illustration of the three-dimensional free surface motion. It is seen that the free surface does not deform appreciably across the channel, and that the shape of the interface is similar to those shown previously for the two-dimensional case. The velocity fields at the symmetry plane and at several cross channel stations are shown

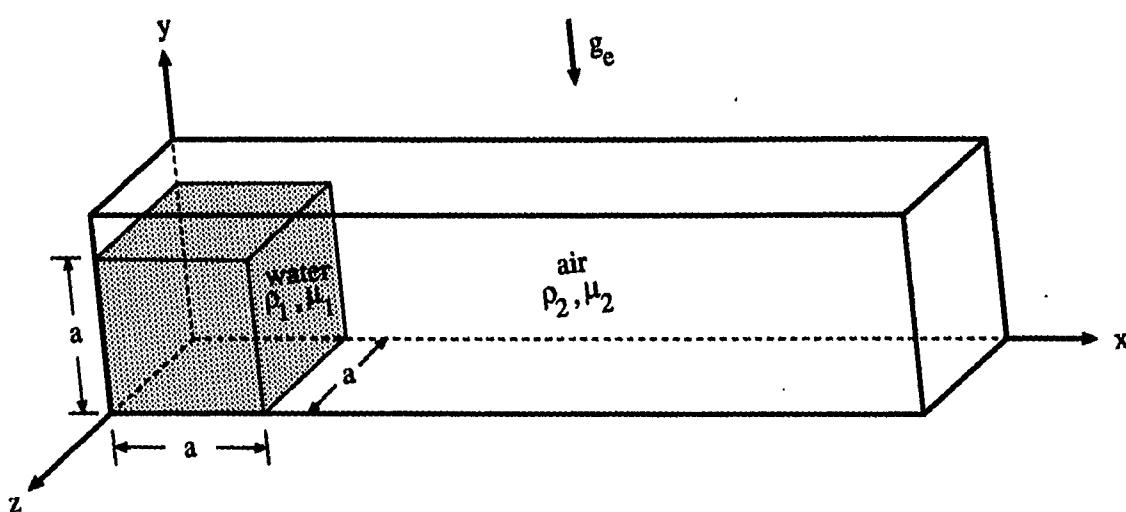


Figure 6.21: Illustration of the three-dimensional broken dam problem

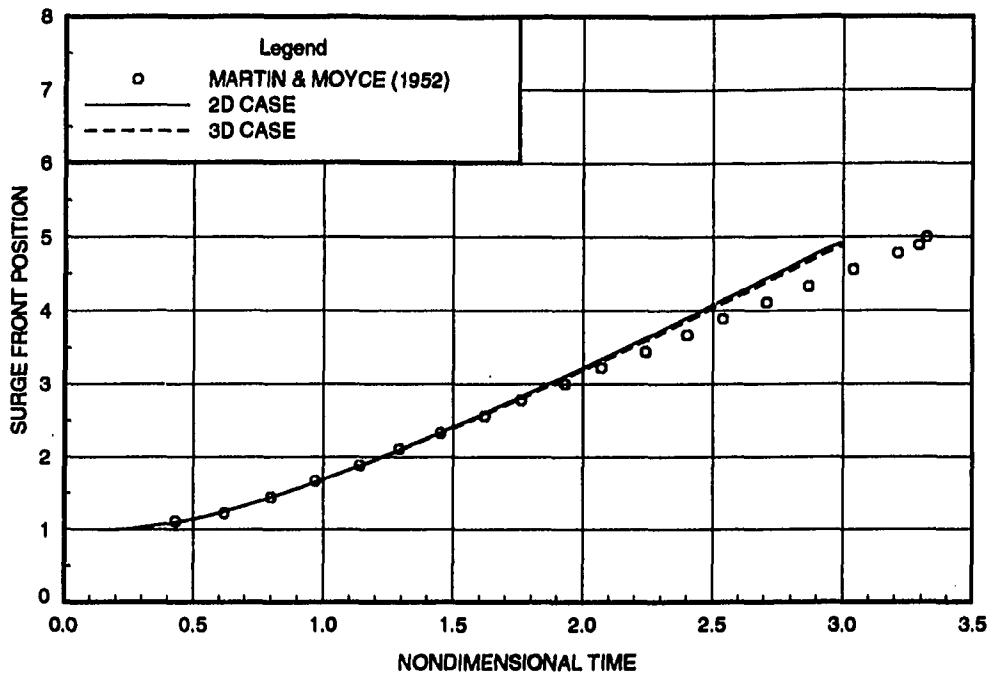


Figure 6.22: Surge front position versus time for $a = 0.05715 \text{ m}$

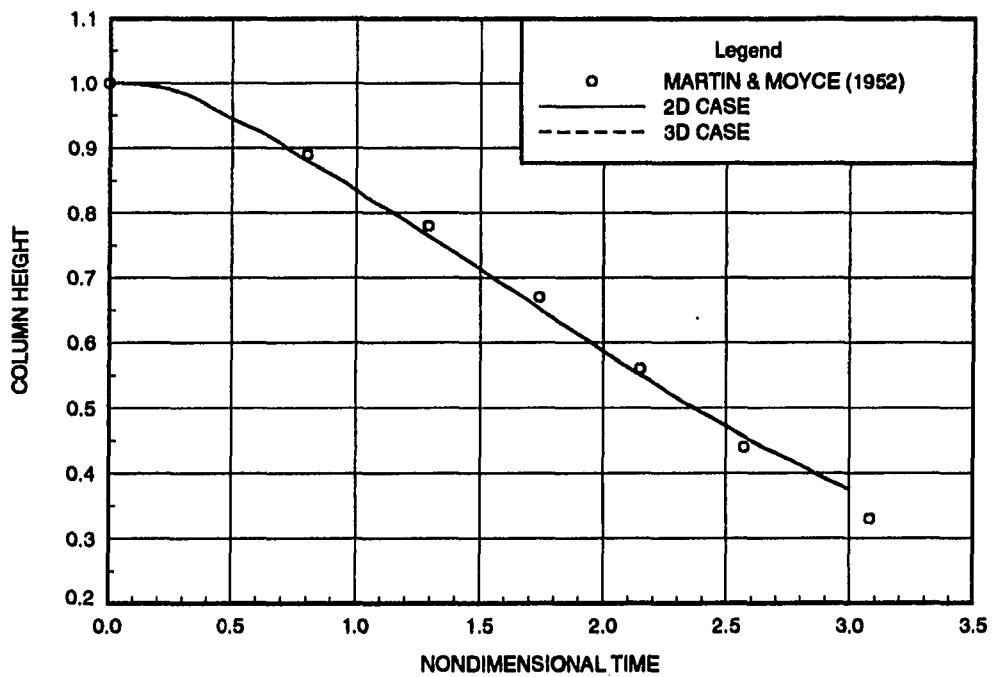
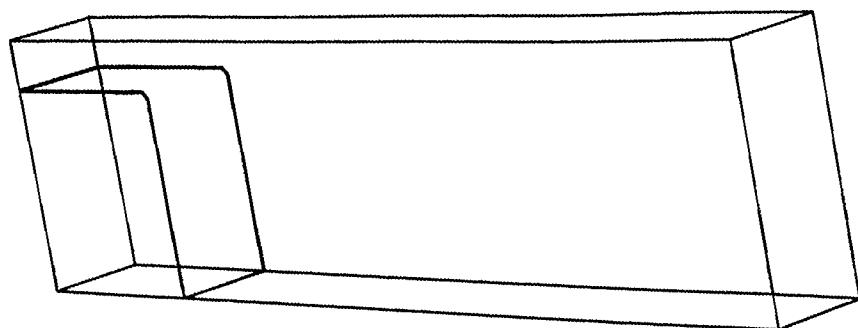


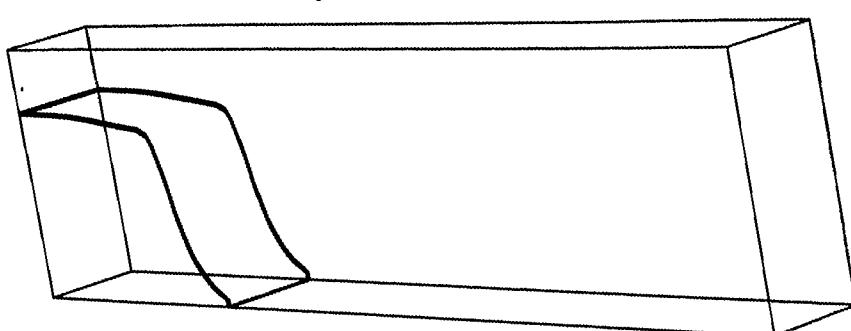
Figure 6.23: Column height versus time for $a = 0.05715 \text{ m}$

146

TIME = 0.0



TIME = 0.6



TIME = 1.2

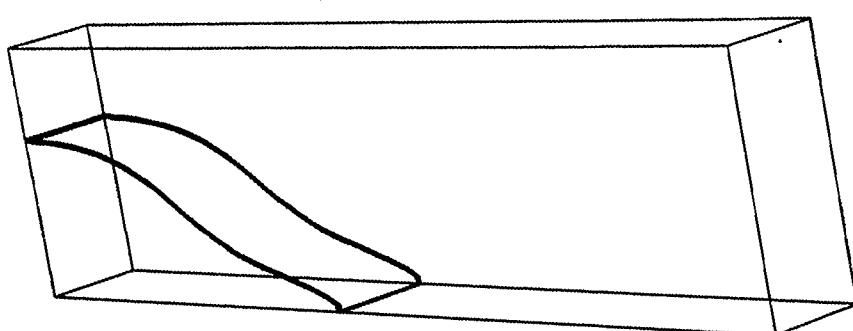
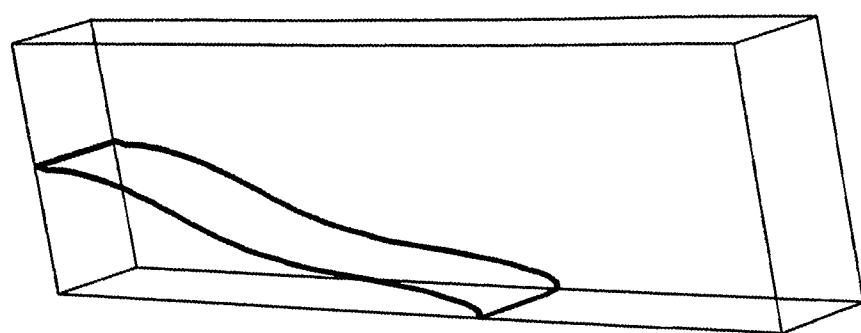


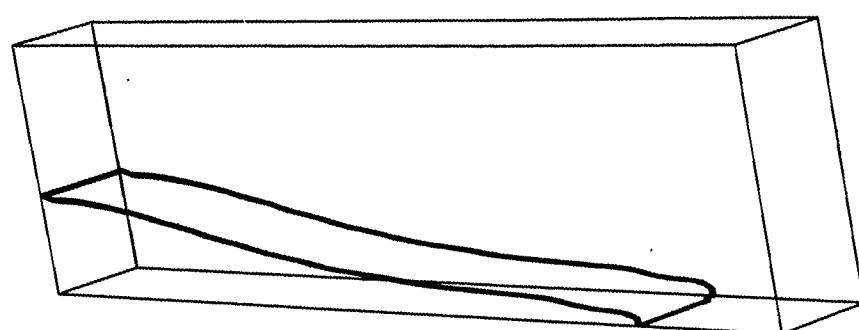
Figure 6.24: Free surface motion for the 3D broken dam problem

147

TIME = 1.8



TIME = 2.4



TIME = 3.0

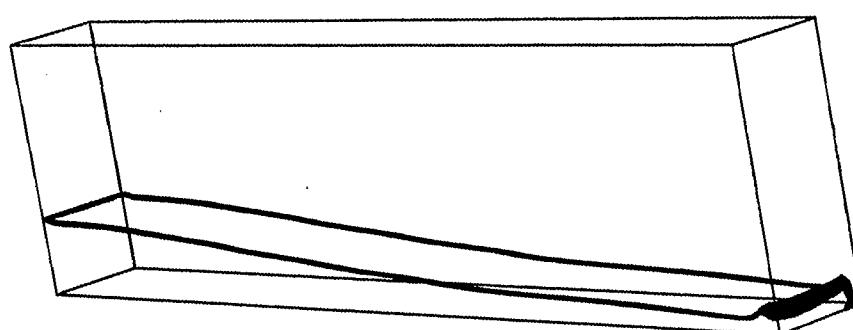


Fig. 6.24: (Continued)

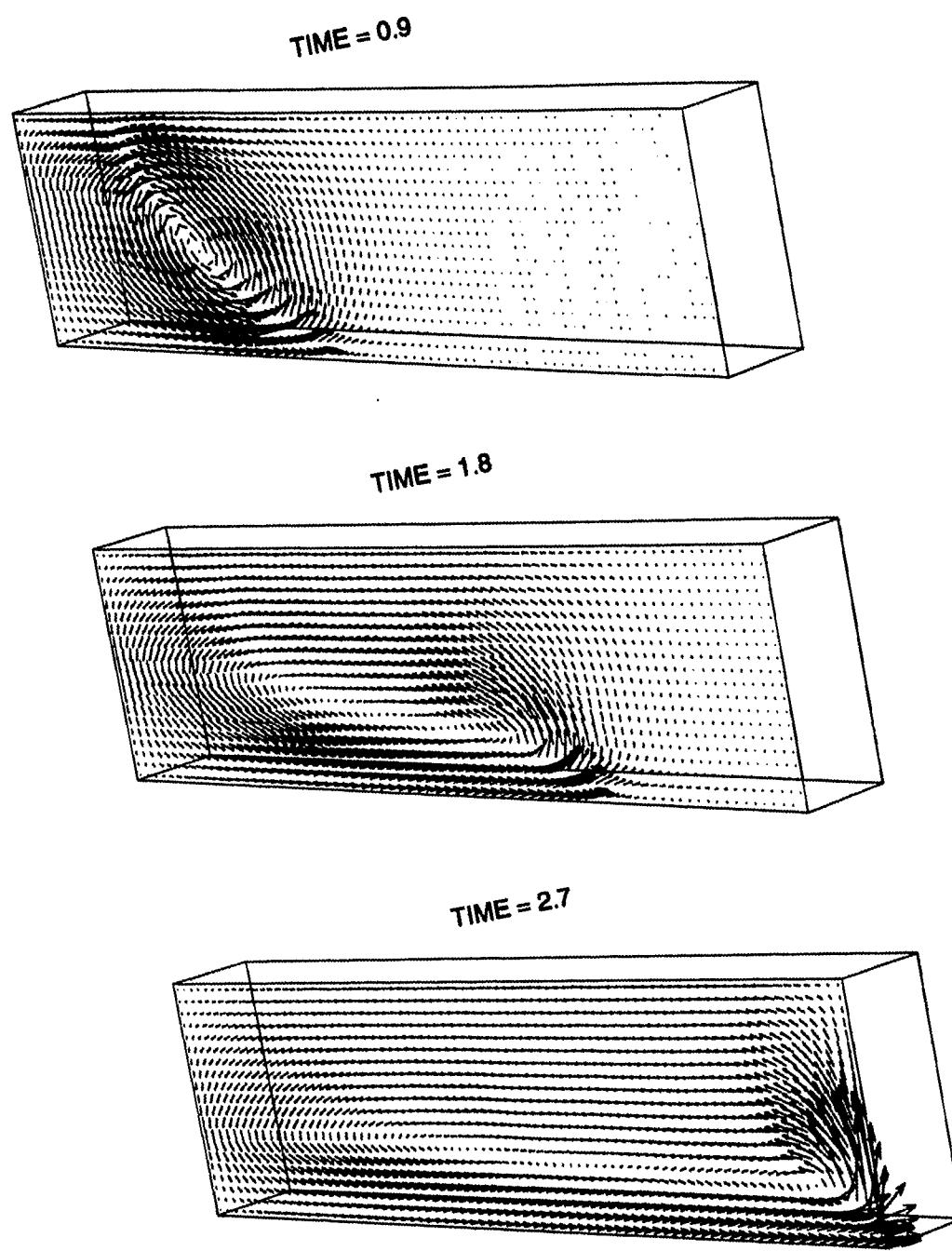


Figure 6.25: Symmetry plane velocity fields for the 3D broken dam problem at selected times

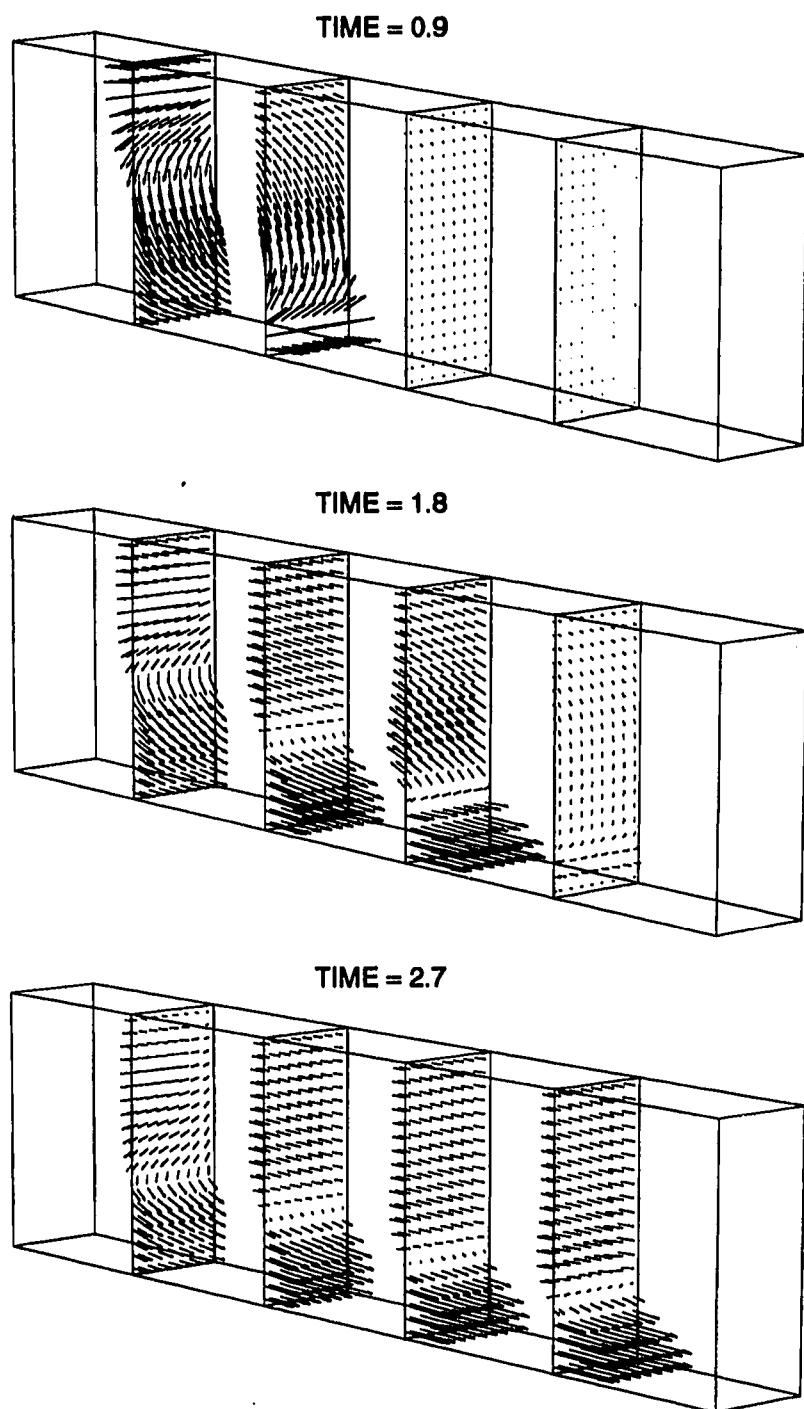


Figure 6.26: Cross channel velocity fields for the 3D broken dam problem at selected times

Table 6.4: Case summary for the three-dimensional broken dam problem

ρ_1 (kg/m ³)	ρ_2 (kg/m ³)	μ_1 (N - s/m ²)	μ_2 (N - s/m ²)	g_e (m/s ²)	β (N/m ²)	Fr	We
1000	1.21	1×10^{-3}	1.81×10^{-5}	9.81	5×10^4	1	∞

Case	Grid	Δt^*	Re
1	$80 \times 20 \times 10$	0.002	42792

in Figs. 6.25 and 6.26. As in the two-dimensional case, a large vortex is observed in the vicinity of the density interface. Also, some three-dimensionality of the flow is observed in the cross channel velocity plots.

6.5 Axisymmetric Spin-up in a Spherical Container

As was mentioned in Chapter 1, the present research was motivated by the particular problem of liquid sloshing within a partially-filled spherical container. It was important, therefore, to examine the effectiveness of the surface capturing approach for this class of problems. This section presents some results for an axisymmetric spin-up of a spherical container half-filled with glycerine. This problem has an easily-calculated steady state solution (see Appendix H) with which the calculations can be compared, and has been used as a validation case in previous studies (Chen, 1990; Kassinos and Prusa, 1990).

The particular problem under consideration is illustrated in Fig. 6.27. An air-filled spherical container of radius R is half-filled with glycerine at standard conditions. At time $t = 0+$, the tank is impulsively spun about its vertical axis at a constant angular velocity, ω_0 , in the presence of a vertical gravitational field.

In order to simplify the problem specification, the initial conditions for the flow-

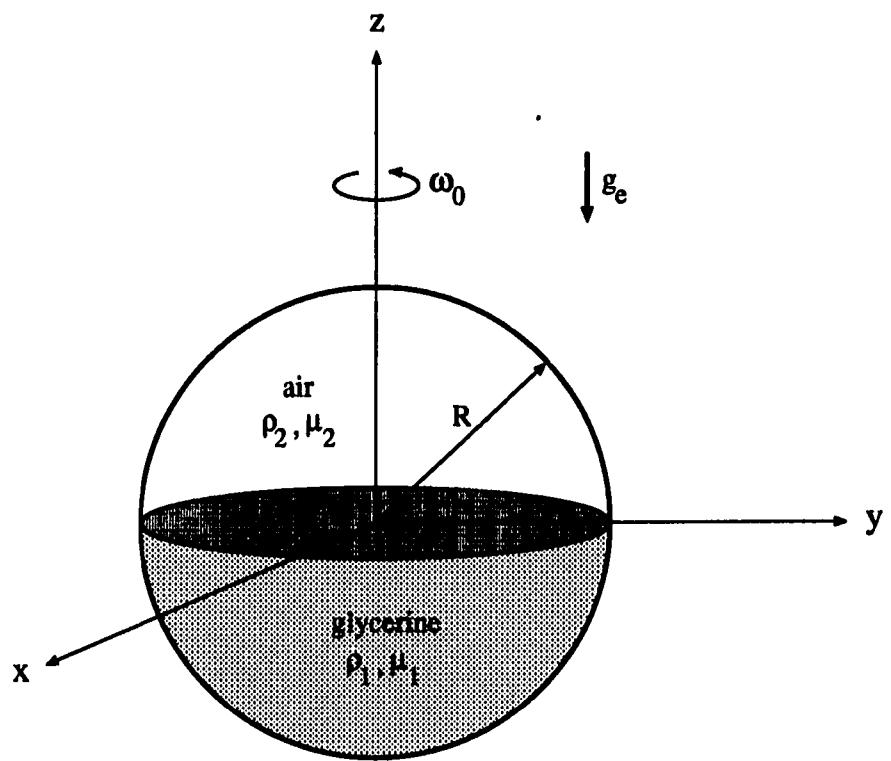


Figure 6.27: Illustration of axisymmetric spin-up problem

field were prescribed using the "initially capped" model. This model assumes that the fluid is brought to solid body rotation under a cap which restrains the motion of the free surface. This cap is then removed at time $t = 0+$, thereby allowing the free surface to seek a new equilibrium position consistent with the gravitational and centripetal accelerations acting on the fluid.

The grid points for this geometry were generated using a bispherical coordinate system. The coordinate transformation relations for this system are given by (Moon and Spencer, 1971)

$$x = \frac{R \cos \xi \sin \zeta}{\cosh \eta - \cos \zeta} \quad (6.4)$$

$$y = \frac{R \sin \xi \sin \zeta}{\cosh \eta - \cos \zeta} \quad (6.5)$$

$$z = \frac{R \sinh \eta}{\cosh \eta - \cos \zeta} \quad (6.6)$$

where $0 \leq \xi \leq 2\pi$, $-\infty < \eta < \infty$, and $\pi/2 \leq \zeta \leq \pi$. This coordinate system is illustrated in Fig. 6.28. It can be seen that the geometry contains metric singularities at the north and south poles and along the vertical symmetry axis. In addition, the periodicity of the circumferential coordinate ξ makes surfaces $\xi = \xi_{min}$ and $\xi = \xi_{max}$ reentrant boundaries.

Since the cell face areas at the metric singularities are zero, the values of the solution variables at these boundaries will not affect the flux balance over the adjacent cells. Therefore, boundary values of density and pressure at the north and south poles were extrapolated from adjacent interior values, while the velocities were set to zero. Along the vertical symmetry axis, the u and v components of velocity were set to

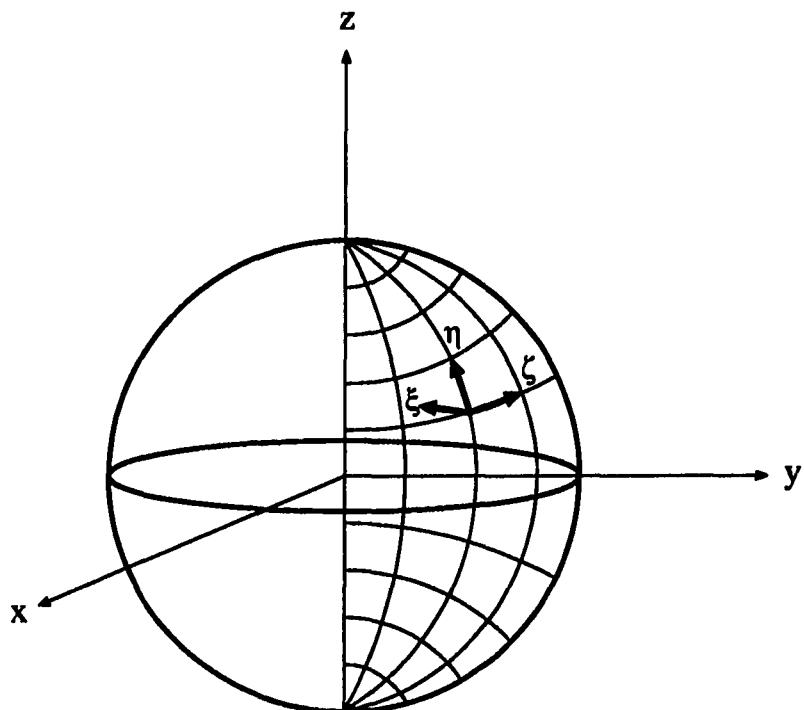


Figure 6.28: The bispherical coordinate system

zero, and the remaining variables obtained from circumferential averages of the values at adjacent cells.

The specific grid employed in the present calculations consisted of $13 \times 32 \times 16$ control volumes in the ξ , η , and ζ coordinate directions, respectively. This grid is illustrated in Fig. 6.29. Although this particular problem is symmetric and, thus, required only a coarse resolution in the circumferential direction, calculations were made using the full three-dimensional algorithm in order to validate the flow solver for this geometry. The initial density field was set as shown in Fig. 6.27, and the initial velocities were set to zero (in accordance with solid body rotation relative to the moving, container-fixed coordinate system). Reference quantities for nondimen-

Table 6.5: Case summary for the axisymmetric spin-up problem

ρ_1 (kg/m ³)	ρ_2 (kg/m ³)	μ_1 (N·s/m ²)	μ_2 (N·s/m ²)	R (m/s ²)	g_e (N/m ²)	β	We
1264	1.21	1.492	1.81×10^{-5}	0.5	9.81	1×10^5	∞

Case	Grid	Δt^*	ω (rpm)	Re	Fr
1	$13 \times 32 \times 16$	0.001	30	665	0.503
2	$13 \times 32 \times 16$	0.001	60	1331	1.006

sionalization were selected as follows: $L_r = R$, $V_r = \omega_0 R$, $\rho_r = \rho_1$, and $\mu_r = \mu_1$.

Two sets of calculations were made, corresponding to angular velocities of 30 rpm and 60 rpm. The key parameters associated with these calculations are summarized in Table 6.5. Due to the computational effort involved, the calculations were only allowed to proceed until the steady-state free surface shape was approached.

The free surface motion and velocity fields for the 30 rpm case are depicted in Figs. 6.30 – 6.31. These plots show two constant ξ surfaces oriented 180 degrees apart in the x-z plane. The figures show the liquid flowing radially outward due to the centripetal acceleration arising from the axisymmetric spin. A vortical circulation pattern is set up in the air in response to the displacement of volume caused by the liquid motion. The free surface gradually evolves into the characteristic parabolic shape, while the magnitude of the fluid velocities decreases as the system begins to approach steady-state.

The computed free surface positions (z/R) at the sphere wall and centerline (vertical axis) for the 30 rpm case are plotted as functions of nondimensional time in Fig. 6.32. Also shown are the corresponding steady-state analytical positions determined using the algorithm presented in Appendix H. It is seen that the computed

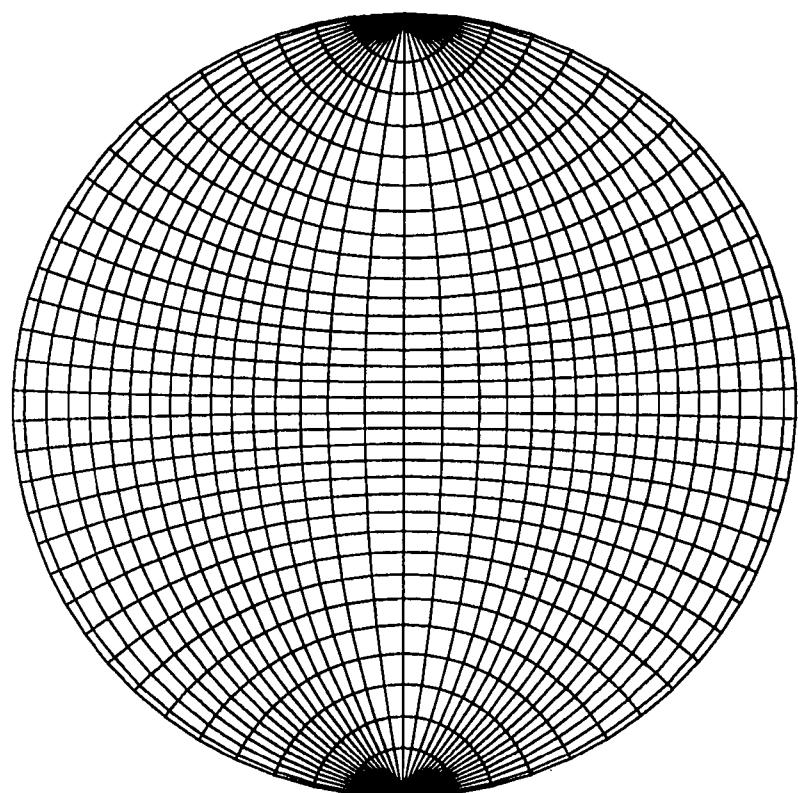


Figure 6.29: Cross sectional view of grid used in axisymmetric spin-up calculations

wall free surface position overshoots the analytical result by about 27% before recovering. The centerline position, on the other hand, has not yet leveled off. Clearly, the calculations would need to be carried out further to determine if the centerline free surface has indeed reached steady-state. The relatively poor level of agreement between the computed free surface positions and the analytical solution can probably be traced to the relatively coarse grid used in the calculations and the inevitable diffusion of the density interface. Some possible remedies for the density interface diffusion problem will be discussed in Chapter 7.

Results for the 60 rpm case are presented in Figs. 6.33 – 6.35. The nondimensional time required to approach steady state was longer than in the 30 rpm case. This is consistent with the relation given by Benton and Clark (1974), which asserts that the nondimensionalized spin-up time is proportional to $\sqrt{\omega_0}$. The features of the flowfield are seen to be similar to those for the 30 rpm case.

The computed free surface positions at the sphere wall and centerline for the 60 rpm case are compared with the analytical solution in Fig. 6.32. As in the 30 rpm case, the computed wall free surface position overshoots the analytical result, this time by about 12%. The centerline free surface position levels off near the end of the transient, and shows fair agreement with the analytical result. Again, the errors in the computed free surface positions are probably due to the coarse grid and the diffusion of the density interface.

6.6 Computational Effort

To provide the reader with an appreciation for the computational effort required by the calculations described in the previous sections, a summary of representative

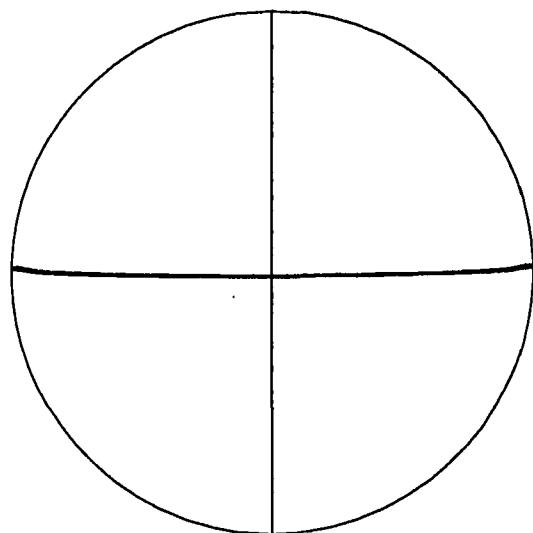
Table 6.6: Representative CPU times (HP 9000/730) for selected validation cases

Problem	CPU time (hr : min : sec)
Broken Dam 2D (80 × 20 grid)	9:48:15
Broken Dam 2D (120 × 30 grid)	18:04:01
Broken Dam 2D (160 × 40 grid)	40:08:59
Rayleigh-Taylor (30 × 90 grid)	18:12:23
Rayleigh-Taylor (40 × 120 grid)	44:29:06
Broken Dam 3D (80 × 20 × 10 grid)	154:47:43
Axisymmetric Spinup (30 rpm - 13 × 32 × 16 grid)	60:43:32
Axisymmetric Spinup (60 rpm - 13 × 32 × 16 grid)	96:37:37

CPU times for selected cases is presented in Table 6.6. These times, which apply exclusively to the HP 9000/730 workstation, range from about 9 hr 48 min for the two-dimensional broken dam problem on the baseline grid to 154 hr 47 min for the three-dimensional broken dam problem. It should be noted that since the flow solver must share its processing time with other jobs, the actual (wall) time required to complete a calculation is invariably longer than the CPU time.

Since no significant effort was made to optimize specific subroutines in the flow solvers, it is likely that the times recorded in Table 6.6 could be improved upon. Indeed, as mentioned in Section 6.1, it would be beneficial to modify each program to permit significant vectorization on supercomputers such as the NASA NAS Crays. The computational resources provided by supercomputers would allow much more challenging two and three-dimensional calculations to be carried out.

TIME = 0.2



TIME = 0.4

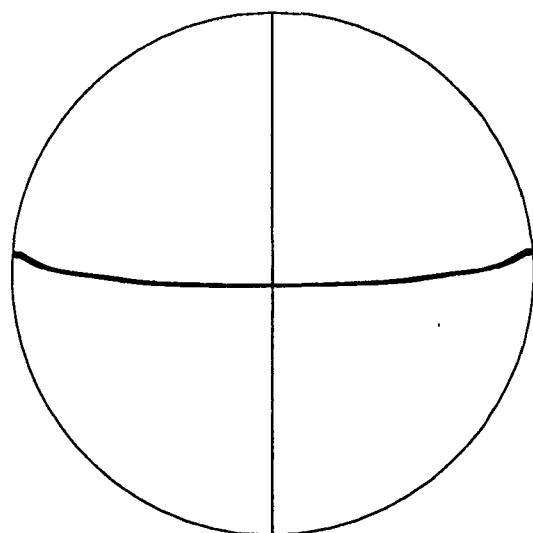
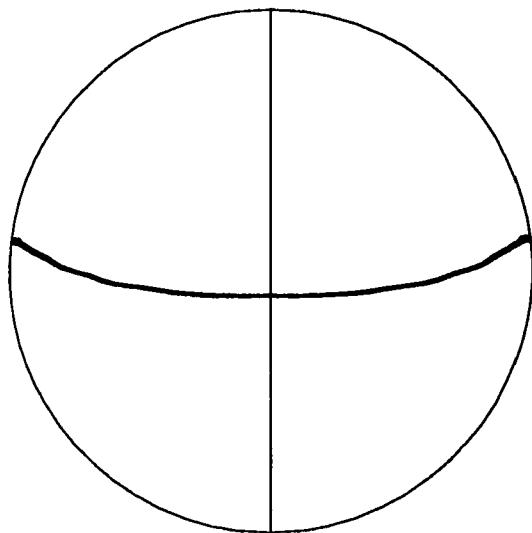


Figure 6.30: Free surface motion for the axisymmetric spin-up problem: 30 rpm case

TIME = 0.6



TIME = 1.0

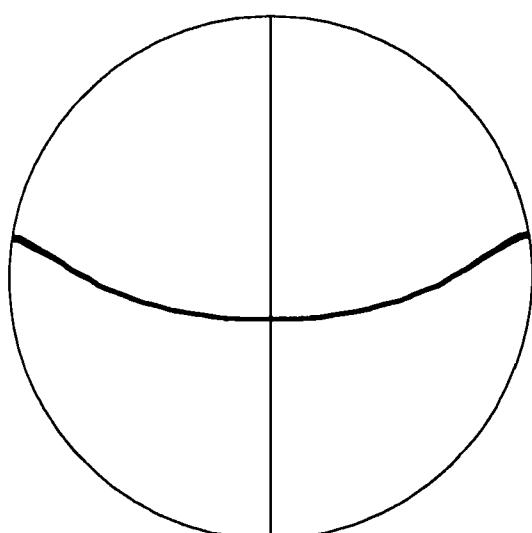
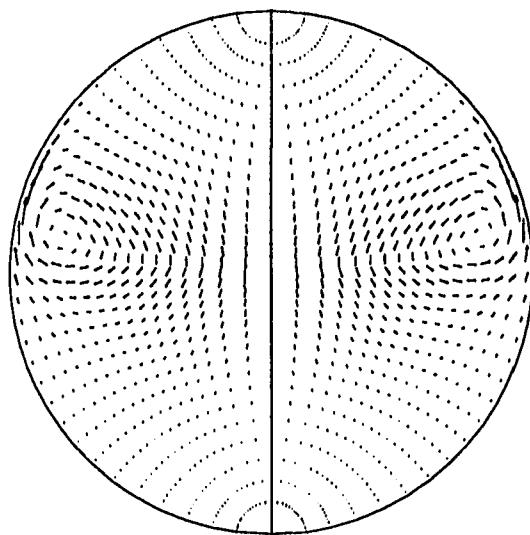


Fig. 6.30: (Continued)

160

TIME = 0.2



TIME = 0.4

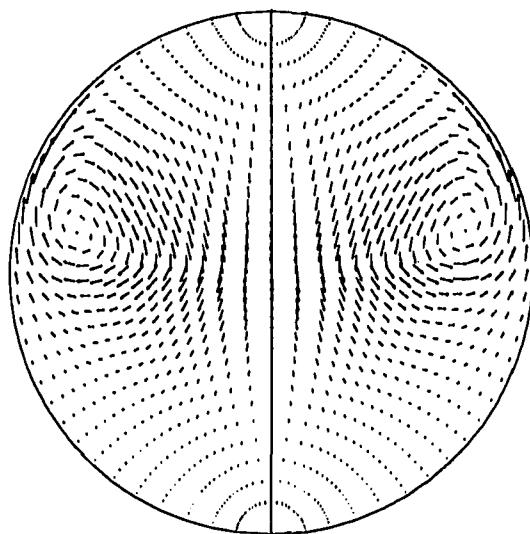
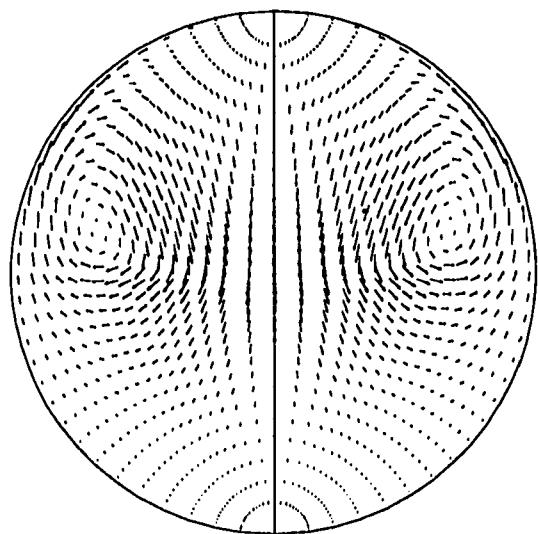


Figure 6.31: Velocity field for the axisymmetric spin-up problem: 30 rpm case

TIME = 0.6



TIME = 1.0

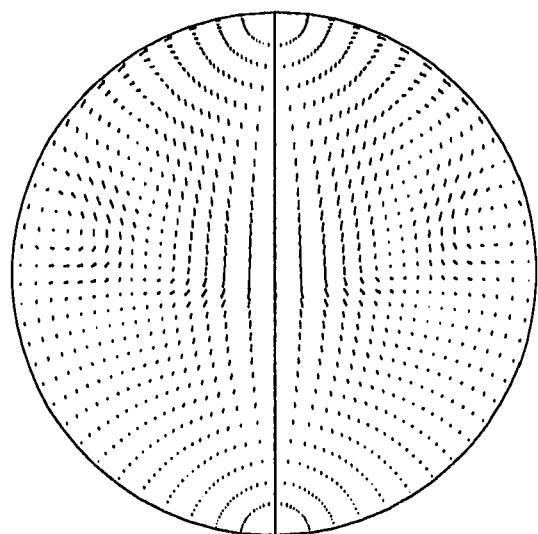


Fig. 6.31: (Continued)

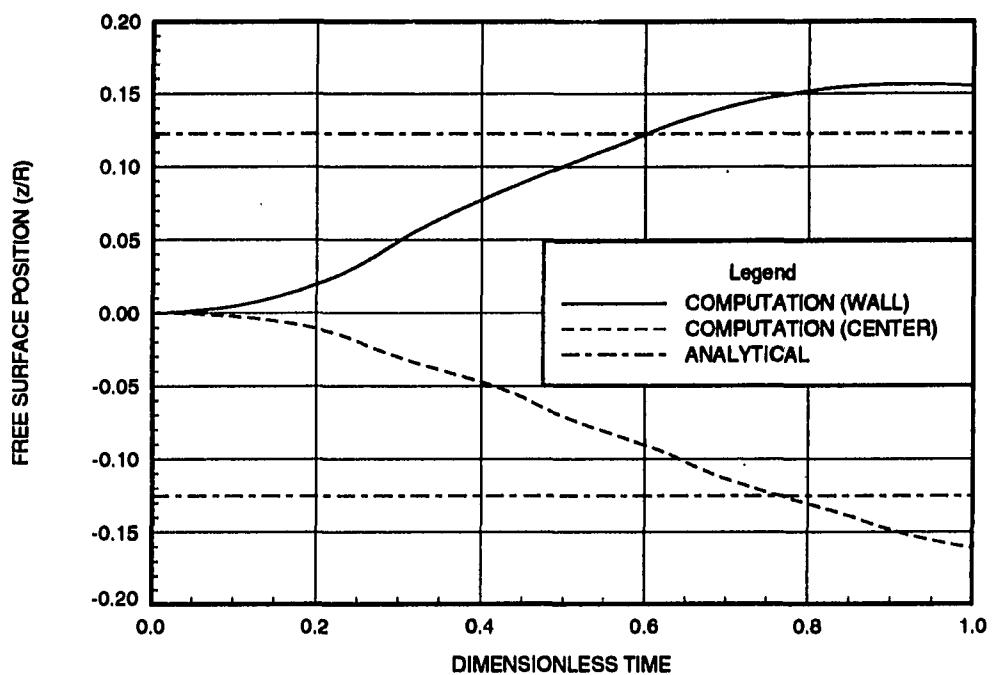
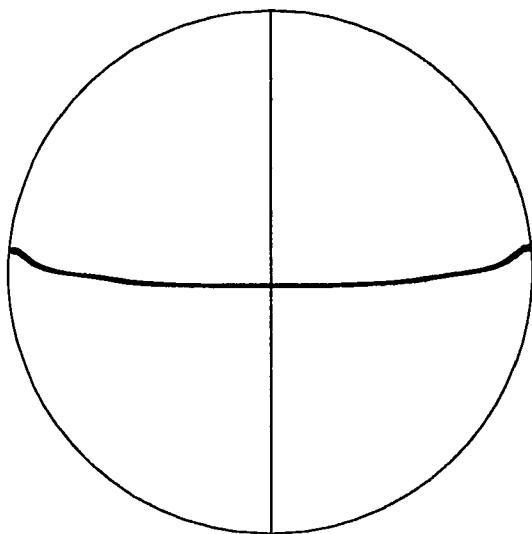


Figure 6.32: Comparison of computed wall free surface position with the steady-state, analytical solution: 30rpm case

TIME = 0.4



TIME = 0.8

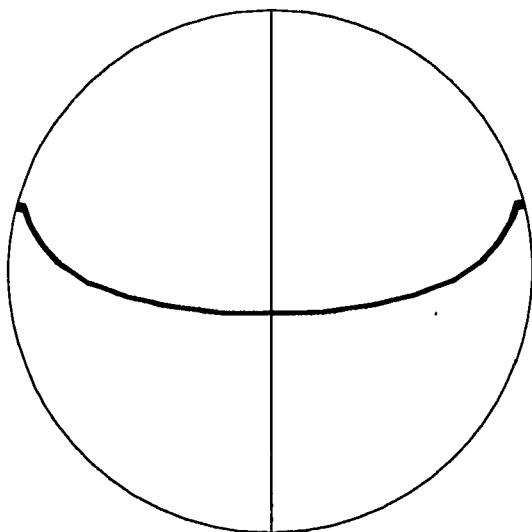
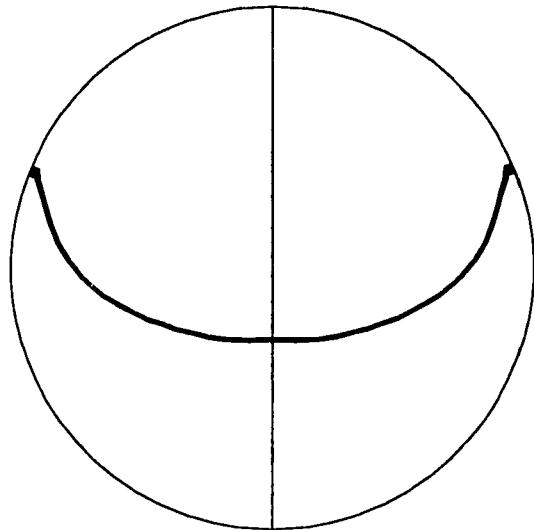


Figure 6.33: Free surface motion for the axisymmetric spin-up problem: 60 rpm case

TIME = 1.2



TIME = 2.0

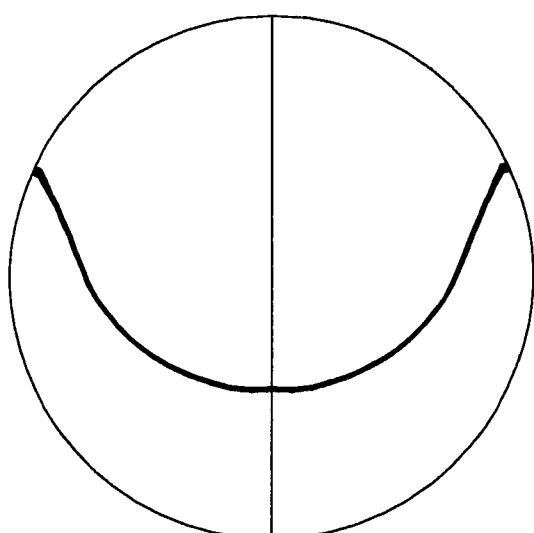


Fig. 6.33 : (Continued)

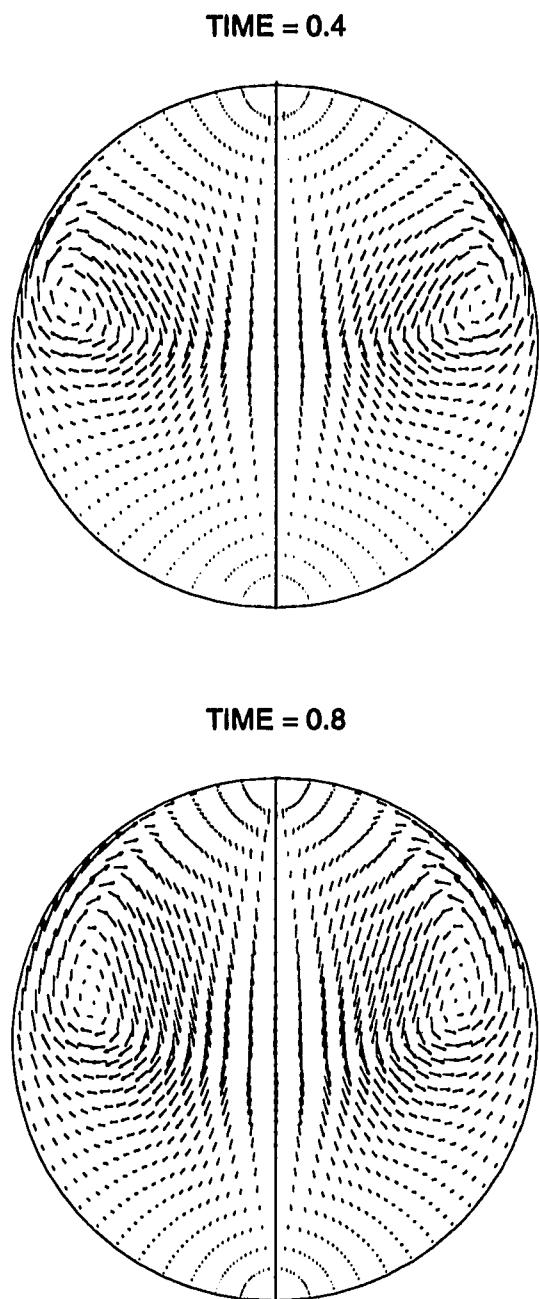
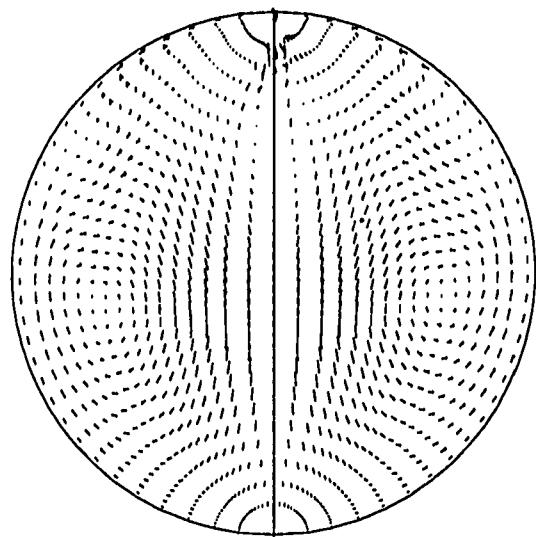


Figure 6.34: Velocity field for the axisymmetric spin-up problem: 60 rpm case

TIME = 1.6



TIME = 2.0

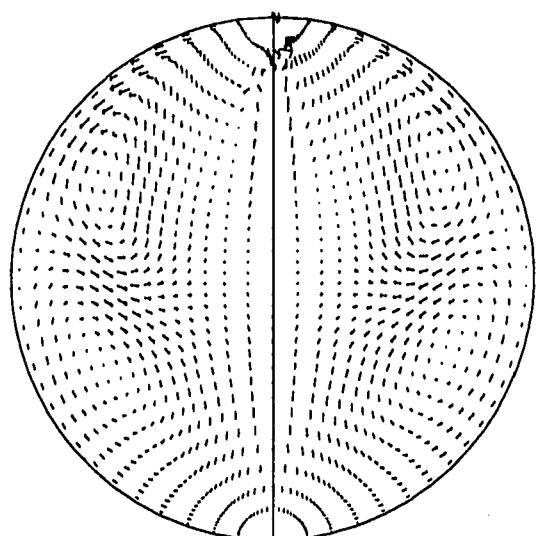


Fig. 6.34: (Continued)

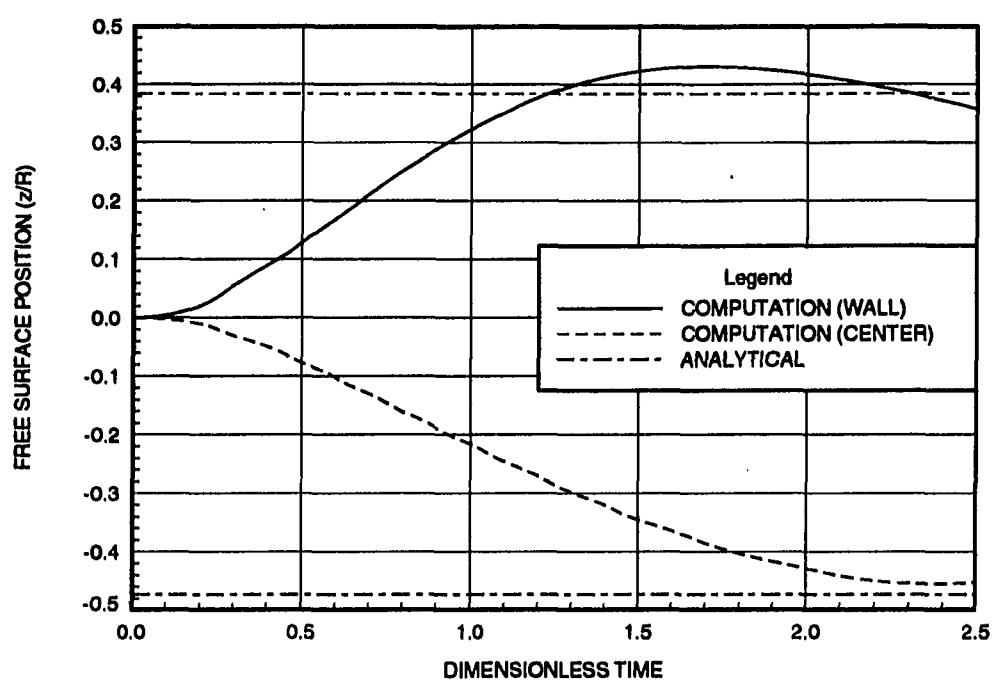


Figure 6.35: Comparison of computed wall free surface position with the steady-state, analytical solution: 60rpm case

7. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

7.1 Summary

The preceding chapters have described the development of a new surface capturing method for computing viscous free surface flows in partially-filled containers. The algorithm employed the artificial compressibility method and a dual time stepping strategy to develop a conservative, implicit, finite volume formulation of the equations of motion for a viscous, incompressible, variable density fluid. The use of a consistent formulation in both the liquid and gas regions permitted the free surface to be automatically captured as a discontinuity in the density, and thereby eliminated the need for special free surface tracking procedures.

The capabilities of the surface capturing method were demonstrated by calculating solutions to several challenging two and three-dimensional problems. The first problem considered was the two-dimensional broken dam problem, using the properties of water for the liquid and air for the gas. An extensive set of calculations were carried out to examine the effects of grid refinement (in both space and time), Reynolds number, upwind-biased interpolants, and limiter functions. The computed solutions were found to be relatively insensitive to grid refinement and Reynolds number. A comparison of computed surge front column height positions as functions of time showed reasonably good agreement with the experimental data of Martin and

Moyce (1952). In addition, the free surface profiles (plotted as density contours about the mean density value) compared favorably with photographs from the Martin and Moyce experiments. The choice of interpolant and limiter functions used in the inviscid flux construction was found to have a significant impact on the level of numerical diffusion at the free surface. Specifically, it was shown that the best solutions (in terms of computed density profiles) were obtained by using the QUICK interpolant for all variables, the compressive minmod limiter for the density, the minmod limiter for pressure, and no limiting on the velocities. This combination of interpolants and limiters was adopted in all subsequent calculations. Total mass conservation checks showed less than 0.01% change over the transient, a result which was replicated in all of calculations performed in this study. Finally, numerical experiments with the broken dam problem revealed that the pseudo-compressibility parameter β^* must be made suitably large (generally on the order of $10^1 - 10^2$) in order to suppress the formation of unphysical oscillations in the density field near the free surface.

A two-dimensional viscous Rayleigh-Taylor instability was examined next. This problem considered two fluid layers with a density ratio of two and a uniform kinematic viscosity. The initial condition for the flow was provided by a sinusoidal velocity perturbation at the interface between the two fluids. Solutions computed for two different Reynolds numbers replicated the characteristic mushroom shape of the interface. The higher Reynolds number case exhibited a more complex interface shape due to the reduced effects of viscosity. A series of calculations were also made to examine the effect of Reynolds number on the linear (early time) growth rate of the perturbation. These results were found to be in good agreement with the analytical results of Chandrasehkar (1961).

The three-dimensional algorithm was tested first on a three-dimensional version of the broken dam problem. Due to the symmetry of the geometry, only half of the channel was employed in the calculation, with a vertical symmetry plane being established at the center of the channel. The computed surge front and column height positions at the symmetry plane were found to agree with both the data of Martin and Moyce (1952) and a previously-computed two-dimensional solution.

The final case that was examined was an axisymmetric spin-up of an air-filled, spherical container half-filled with glycerine. Two spin-up speeds (30 and 60 rpm) were considered using the "initially capped" model to prescribe the initial conditions. The resulting flowfields were found to be qualitatively correct, showing the gradual evolution of the free surface towards the characteristic parabolic shape. Overall, the computed free surface positions on the sphere wall and centerline, showed only fair agreement with the predicted analytical solutions. The errors were attributed to both a lack of resolution in the grid employed in the calculations and the diffusion of the density interface.

7.2 Conclusions

In view of the goals set forth at the beginning of this dissertation, it can be stated that the present research has successfully demonstrated the viability of the surface capturing approach in computing complex free surface flows. As compared the MAC, VOF, and previous surface capturing approaches, the present method possesses a number of desirable features. First, the formulation is fully coupled, conservative, implicit, and formally second order accurate in space and time. This contrasts with the first order (in time), hybrid explicit-implicit schemes employed in

most MAC and VOF formulations. Second, the consistent formulation of the inviscid fluxes (using slope-limited, upwind-biased interpolants) permits easy extension of the method to three-dimensional geometries and avoids the need for ad hoc procedures to prevent numerical diffusion of the free surface. Third, the method is capable of computing two fluid flows for a range of density ratios, and, unlike the MAC and VOF methods, can compute the flowfields in both fluids simultaneously.

Another noteworthy accomplishment of the present research is the use of the artificial compressibility method to compute a viscous, incompressible two-fluid flow. The resulting formulation employs the Jacobian inviscid flux vector (Eq. 4.32) and its associated eigensystem, the derivation of which is thoroughly documented in the Appendix. To the author's knowledge, this formulation has not been examined previously (particularly in three dimensions), although it is possible that some similar work has been done but not extensively reported.

The experience gained from the calculations also revealed several weaknesses of the current method. The most important of these, in the author's opinion, is the gradual diffusion of the density interface over time. Given the manner in which the numerical fluxes are computed (i.e. using one-dimensional upwind formulas), some numerical diffusion of the density interface was unavoidable. While the use of compressive, slope-limited, upwind-biased interpolants prevents excessive smearing in the short run, eventually the growth of the diffusion zone about the interface leads to numerical difficulties, which, in certain cases, also leads to divergence. Two ways of combating this problem include incorporating a multidimensional upwinding scheme into the formulation, and employing solution-adaptive gridding techniques. These topics will be discussed further in the next section.

Another issue which needs to be addressed concerns the optimal value of the pseudo-compressibility parameter, β^* . The present research established that a sufficiently “large” value of β^* was necessary to prevent unphysical overshoots in the density near the free surface. However, a value that is too large also leads to slow convergence (or even divergence) of the pseudo-time marching process. Therefore, some “optimal” value of β^* must exist, although it is likely that this value is problem dependent. A better understanding of the role of β^* in the formulation could lead to a more objective means of prescribing its value.

7.3 Recommendations for Further Research

The present research has shown that a discontinuity capturing approach can be applied to free surface flows. This idea can clearly be developed further, both by introducing modifications which correct some of the shortcomings of the present method, and by extending the methodology to handle other flow regimes. The subsections below discuss several topics which could be examined in future research on surface capturing methods.

7.3.1 Multidimensional upwind schemes

As was mentioned in Chapter 4, the use of one dimensional MUSCL formulas in a multidimensional scheme is inherently flawed since it assumes that information travels along waves which propagate in directions normal to the cell faces. This practice introduces numerical diffusion into the computed solution, which, in the case of the density field, results in a smearing of the free surface and an attendant loss of accuracy.

One way to reduce this problem is to incorporate a multidimensional form of the MUSCL scheme into the formulation. Multidimensional schemes have received considerable attention in recent years, and a lucid discussion of the state of the art can be found in articles by P.L. Roe (1993) and H. Deconinck (1993). In the present case, the simplest approach would be to introduce a multidimensional interpolant which constructs \mathbf{U}^L and \mathbf{U}^R according to the direction of the local velocity vector at a cell face. While this scheme would lead to a larger computational stencil and, in turn, a more complicated flux evaluation and linearization, the potential for greatly enhanced accuracy and robustness (especially for three-dimensional flows) would make the effort worthwhile.

7.3.2 Solution-adaptive grids

Another method of controlling numerical diffusion at the free surface would be to employ a solution-adaptive grid. Solution adaptive grids have been used to resolve shock waves, shear layers, and other discontinuities in compressible flows. For structured grids such as those employed in the present work, the most natural form of adaptive gridding would be to use a system of nested grids. The basic algorithm would involve scanning the numerical solution for high gradient regions, marking cells whose gradients satisfy some threshold criterion, and finally subdividing the marked cells. A discussion of some advanced solution adaptation techniques for structured grids can be found in Evans et al. (1991).

7.3.3 Surface tension

It was emphasized in Chapter 2 that surface tension can have an important influence on the fluid motion in many classes of problems. Some examples include capillary flows, bubble and droplet dynamics, and microgravity flows. It would clearly be desirable to incorporate some model of surface tension into the present formulation. One possible approach, which is discussed in both Chapter 2 and Appendix A, is based on the ideas of Kataoka (1986) and is similar in form to the procedure discussed by Brackbill, Kothe, and Zemach (1992). It should be noted that this procedure is uncoupled from the flowfield solution in that the surface tension force is computed using the density field at the current pseudo-time level. This force is then fixed during the flowfield solution step. Since the procedure for computing the surface tension force is nontrivial, one may wish to employ an even looser coupling by only computing the surface tension at the beginning of the physical time step.

7.3.4 Turbulence modeling

All of the problems considered in the present work have assumed laminar flow. While this assumption is valid for sufficiently low Reynolds and relatively simple free surface motions, it is possible for turbulent conditions to be established at higher Reynolds numbers and large amplitude free surface motions.

The calculation of turbulence for single phase incompressible flows is still an active area of research. Needless to say, the application of some form of turbulence modeling to the present two-fluid system would introduce many complications and uncertainties. Nevertheless, since the present formulation is constructed in a consistent way, it should be possible to develop both simple algebraic mixing length models

and more complicated one or two equation models which are similar to their single phase counterparts.

7.3.5 Two phase flows

The present surface capturing method can be viewed as a simplified model for two phase flows wherein both phases are immiscible, incompressible, and isothermal. While this model may be adequate for many purposes, it would be desirable to introduce limited compressibility effects into the gas phase. This can be accomplished by modifying the pseudo-compressibility equation as follows (Hirt and Nichols, 1980):

$$\frac{1}{\rho c^2} \frac{\partial p}{\partial t} + \frac{1}{\beta} \frac{\partial p}{\partial \tau} + \nabla \cdot \vec{V} = 0 \quad (7.1)$$

Here, c^2 is the adiabatic speed of sound for the gas.

Many of the techniques developed in this dissertation could also be applied to a more general two phase flow model. For example, the local instant two fluid formulation of Kataoka (1986) is given by the following system of equations:

- Conservation of Mass

$$\frac{\partial}{\partial t} (\phi_k \rho_k) + \nabla \cdot \phi_k \rho_k \vec{V}_k = -\rho_{ki} (\vec{V}_{ki} - \vec{V}_i) \cdot \hat{n}_{ki} a_i \quad (7.2)$$

- Conservation of Momentum

$$\begin{aligned} & \frac{\partial}{\partial t} (\phi_k \rho_k \vec{V}_k) + \nabla \cdot [\phi_k (\rho_k \vec{V}_k \vec{V}_k + \bar{\bar{I}} p_k - \bar{\bar{\tau}}_k)] \\ &= \phi_k \rho_k \vec{B}_k + \{-[\rho_{ki} \vec{V}_{ki} \cdot (\vec{V}_{ki} - \vec{V}_i) + p] \bar{\bar{I}} - \bar{\bar{\tau}}_k\} \cdot \hat{n}_{ki} a_i \end{aligned} \quad (7.3)$$

- Conservation of Energy

$$\begin{aligned}
 & \frac{\partial}{\partial t} (\phi_k \rho_k e_{tk}) + \nabla \cdot \left[\phi_k \rho_k \vec{V}_k \left(e_{tk} + \frac{p_k}{\rho_k} \right) \right] \\
 = & \nabla \cdot [\phi_k (-\vec{q}_k + \vec{\tau}_k \cdot \vec{V}_k)] + \phi_k \rho_k \vec{B}_k \cdot \vec{V}_k + \phi_k Q_k \\
 + & [\rho_{ki} e_{tki} (\vec{V}_{ki} - \vec{V} - \vec{q}_{ki} - p_{ki} \vec{V}_{ki} + \vec{\tau}_{ki} \cdot \vec{V}_{ki})] \cdot \hat{n}_{ki} a_i
 \end{aligned} \quad (7.4)$$

- Interface Motion

$$\frac{\partial \phi_k}{\partial t} + \vec{V}_k \cdot \nabla \phi_k = 0 \quad (7.5)$$

In the above, $k = 1, 2$ denotes the individual phases (liquid and gas) of the two phase flowfield, and ϕ_k is a characteristic function which is one in phase k and zero elsewhere. Since only one phase can occupy a point in space at a given point in time, it follows that

$$\phi_1 + \phi_2 = 1 \quad (7.6)$$

Thus, the interface motion can be deduced from a single equation of the form given by Eq. (7.5). The entire system consists now of eleven partial differential equations for the unknowns ϕ_k , ρ_k , \vec{V}_k , and e_{tk} . Additional relations (e.g. an equation of state) are required to obtain closure.

Given the wide range of flow regimes which are possible, Eqs. (7.2) – (7.6) are rarely solved in the form presented above. Instead, practical models are developed by applying various temporal and spatial averaging techniques to the equations, and

supplying appropriate empiricisms to account for subscale interfacial phenomena. A detailed discussion of the two-fluid model can be found in Ishii (1975).

BIBLIOGRAPHY

- Abramson, H.N. (1966). "The Dynamic Behavior of Liquids in Moving Containers," Southwest Research Institute, NASA SP-106.
- Anderson, M.D. (1988). "Instrumentation of a spin-stabilized spacecraft simulator with liquid fuel stores," M.S. thesis, Iowa State University, Ames, Iowa.
- Anderson, D.A., Tannehill, J.C. and Pletcher, R.H. (1984). *Computational Fluid Mechanics and Heat Transfer*, McGraw-Hill Book Co., New York.
- Antontsev, S.N, Kazhikhov, A.V., and Monakhov, V.N. (1990). *Boundary Value Problems in Mechanics of Nonhomogeneous Fluids*, North-Holland, Amsterdam.
- Aris, R. (1962). *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*, Prentice-Hall, Englewood Cliffs, NJ.
- Barth, T.J. (1987). "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," AIAA Paper 87-0595.
- Baumgarten, J.R., Flugrad, D.R., and Prusa, J.M. (1989). "Investigation of Liquid Sloshing in Spin-Stabilized Satellites," ISU-ERI-Ames-90401, Iowa State University.
- Baumgarten, J.R., Flugrad, D.R., and Pletcher, R.H. (1990). "Investigation of Liquid Sloshing in Spin-Stabilized Satellites," ISU-ERI-Ames-90410, Iowa State University.
- Baumgarten, J.R., Flugrad, D.R., and Pletcher, R.H. (1991). "Investigation of Liquid Sloshing in Spin-Stabilized Satellites," ISU-ERI-Ames-92400, Iowa State University.

- Beam, R., and Warming, R.F. (1976). "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *Journal of Computational Physics*, Vol. 22, pp. 87-110.
- Benton, E.R., and Clark, A.C. (1974). "Spin-up," *Annual Reviews of Fluid Mechanics*, Vol. 6, pp. 257-280.
- Brackbill, J.U., Kothe, D.B., and Zemach, C. (1992), "A Continuum Method for Modeling Surface Tension," *Journal of Computational Physics*, Vol. 100, pp. 335-354.
- Briley, W.R., and MacDonald, H. (1973). "Solution of the Three-dimensional Compressible Navier-Stokes Equations by an Implicit Technique," in *Lecture Notes in Physics*, Vol. 35., Springer-Verlag, New York, pp. 105-110.
- Buratynski, E.K., and Caughey, D.A. (1986). "An Implicit LU Scheme for the Euler Equations Applied to Arbitrary Cascades," *AIAA Journal*, Vol. 24, No. 1, pp. 39-46.
- Chan, K.C., and Street, R.L. (1970). "A Computer Study of Finite-Amplitude Water Waves," *Journal of Computational Physics*, Vol. 6, pp. 68-94.
- Chandrasekhar, S. (1961). *Hydrodynamics and Hydromagnetic Stability*, Oxford at the Clarendon Press, Oxford.
- Chen, K.-H. (1990). "A primitive variable, strongly implicit calculation procedure for two and three-dimensional unsteady viscous flows: application to compressible and incompressible flows including flows with free surfaces," Ph.D. dissertation, Iowa State University, Ames, Iowa.
- Chen, K., and R.H. Pletcher (1991). "Simulation of Three-Dimensional Liquid Sloshing Using a Strongly Implicit Calculation Procedure," AIAA Paper 91-1661.
- Chen, K., Kelecy, F.J. and Pletcher, R.H. (1992). "A Numerical and Experimental Study of Three-Dimensional Liquid Sloshing in a Rotating Spherical Container," AIAA Paper 92-0829.
- Chorin, A.J. (1967). "A Numerical Method for Solving Incompressible Viscous Flow Problems, *Journal of Computational Physics*, Vol. 2, pp. 12-26.

- Cooper, R.M. (1960). "Dynamics of Liquids in Moving Containers," *ARS Journal*, Vol. 30, No. 8, pp. 725-729.
- Cowles, D.S. (1987). "Design of a spin-stabilized spacecraft simulator with liquid fuel stores," M.S. thesis, Iowa State University, Ames, Iowa.
- Daily, J.W., and Harleman, D.R.F. (1966). *Fluid Dynamics*, Addison-Wesley Publishing Co., Inc., Reading.
- Daly, B.J. (1967). "Numerical Study of Two Fluid Rayleigh-Taylor Instability," *The Physics of Fluids*, Vol. 10, No. 2, pp. 297-307.
- Deconinck, H. (1993). "Beyond the Riemann Problem, Part II," in *Algorithmic Trends in Computational Fluid Dynamics*, M.Y. Hussani, A. Kumar, and M.D. Salas (Eds.), Springer-Verlag, New York, pp. 369-396.
- Dussan V., E.B. (1983). "The Moving Contact Line," in *Waves on fluid interfaces*, R.E. Meyer, ed., Academic Press, New York, pp. 303-323.
- Edwards, D.A., Howard, B., and Wasan, D.T. (1991). *Interfacial Transport Processes and Rheology*, Butterworth-Heinemann, Boston.
- Evans, A., Marchant, M.J., Szmelter, J., and Weatherill, N.P. (1991). "Adaptivity for compressible flow computations using point embedding on 2-D structured multiblock meshes," *International Journal for Numerical Methods in Engineering*, Vol. 32, pp. 896-919.
- Floryan, J.M. and Rasmussen, H. (1989). "Numerical methods for viscous flows with moving boundaries," *Applied Mechanics Reviews*, Vol. 42, No. 12, pp. 323-340.
- Godunov, S.K. (1959). "A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Mat. Sb.*, Vol. 47, pp. 357-393.
- Harlow, F.H., and Welch, J.E. (1965). "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *The Physics of Fluids*, Vol. 8, No. 12, pp. 2182-2189.
- Harlow, F.H., and Welch, J.E. (1966). "Numerical Study of Large-Amplitude Free-Surface Motions," *The Physics of Fluids*, Vol. 9, No. 3, pp. 842-851.

- Hirsch, C. (1988). *Numerical Computation of Internal and External Flows, Vol. 1*, John Wiley & Sons, New York.
- Hirsch, C. (1990). *Numerical Computation of Internal and External Flows, Vol. 2*, John Wiley & Sons., New York.
- Hirt, C.W., and Shannon, J.P. (1968). "Free Surface Stress Conditions for Incompressible-Flow Calculations," *Journal of Computational Physics*, Vol. 2, pp. 403-411.
- Hirt, C.W., Cook, J.L., and Butler, T.D. (1970). "A Lagrangian Method for Calculating the Dynamics of an Incompressible Fluid with Free Surface," *Journal of Computational Physics*, Vol. 5, pp. 103-124.
- Hirt, C.W., and Nichols, B.D. (1980). "Adding Limited Compressibility to Incompressible Hydrocodes," *Journal of Computational Physics*, Vol. 34, pp. 390-400.
- Hirt, C.W., and Nichols, B.D. (1981). "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," *Journal of Computational Physics*, Vol. 39, pp. 201-225.
- Ishii, M. (1975). *Thermo-fluid Dynamic Theory of Two-Phase Flow*, Eyrolles, Paris.
- Jameson, A., and Turkel, E. (1981). "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, No. 156, pp. 385-397.
- Jameson, A., and Yoon, S. (1987). "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal*, Vol. 25, No. 7, pp. 929-935.
- Jeffrey, A. (1964). "A Note on the Derivation of the Discontinuity Conditions Across Contact Discontinuities, Shocks, and Phase Fronts," *Zeitschrift Fur Angewandte Mathematik und Physik*, Vol. 15, pp. 68-71.
- Jeffrey, A. (1965). "A Note on the Integral Form of the Fluid Dynamic Conservation Equations Relative to an Arbitrarily Moving Volume," *Zeitschrift Fur Angewandte Mathematik und Physik*, Vol. 16, pp. 835-837.

- Jeffrey, A. (1976). *Quasilinear hyperbolic systems and waves*, Pitman Publishing, London.
- Jun, L. and Spalding, D.B. (1988). "Numerical Simulation of Flows with Moving Interfaces," *PhysicoChemical Hydrodynamics*, Vol. 10, No. 5/6, pp. 625-637.
- Kassinos, A.C., and Prusa, J.M. (1990). "A Numerical Model for 3-D Viscous Sloshing in Moving Containers," in *Symposium on Recent Advances and Applications in CFD*, Proceedings of the ASME Winter Annual Meeting, pp. 75-86.
- Kataoka, I. (1986). "Local Instant Formulation of Two-Phase Flow," *International Journal of Multiphase Flow*, Vol. 12, No. 5, pp. 745-758.
- Kordulla, W., and Vinokur, M. (1983). "Efficient Computation of Volume in Flow Predictions," *AIAA Journal*, Vol. 21., No. 6., pp. 917-918.
- Lax, P.D. (1954). "Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation," *Communications in Pure and Applied Mathematics*, Vol. 7, pp. 159-193.
- Leonard, B.P. (1979). "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, pp. 59-98.
- Lewis, D.J. (1950). "The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. II," *Proceedings of the Royal Society of London*, Ser. A, Vol. 202, pp. 81-96.
- Mader, C.L. (1988). *Numerical Modeling of Water Waves*, University of California Press, Berkeley.
- Martin, J.C, and Moyce, W.J. (1952). "An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane," *Philosophical Transactions of the Royal Society of London*, Ser. A, Vol. 244, pp. 312-324.
- Maxwell, T.T. (1977). "Numerical Modelling of Free Surface Flows," Ph.D. dissertation, Imperial College, University of London, London, UK.

- Maxwell, T.T. and Spalding, D.B. (1987). "Numerical Modelling of Free Surface Flows," in *Numerical Methods for Transient and Coupled Problems*, R.W. Lewis, E. Hinton, and B.A. Schrefler, eds., John Wiley & Sons, Ltd, New York, pp. 237-259.
- Merkle, C.L., and Athavale, M. (1987). "Time-Accurate Unsteady Incompressible Flow Algorithms Based On Artificial Compressibility," AIAA Paper 87-1137.
- Moon, P.H., and Spencer, D.E. (1971). *Field theory handbook; including coordinate systems, differential equations and their solutions*, Springer-Verlag, Berlin.
- Mulder, W., Osher, S., and Sethian, J.A. (1992). "Computing Interface Motion in Compressible Gas Dynamics," *Journal of Computational Physics*, Vol. 100, pp. 209-228.
- Nichols, B.D., and Hirt, C.W. (1973). "Calculating Three-Dimensional Free Surface Flows in the Vicinity of Submerged and Exposed Structures," *Journal of Computational Physics*, Vol. 12, pp. 234-246.
- Obermaier, L.A. (1988). "Computer simulation of a spin-stabilized spacecraft simulator with liquid fuel stores," M.S. thesis, Iowa State University, Ames, Iowa.
- Pan, D., and Lomax, H. (1988). "A New Approximate LU Factorization Scheme for the Reynolds-Averaged Navier-Stokes Equations," *AIAA Journal*, Vol. 26, No. 2, pp. 163-171.
- Pan, D., and Chakravarthy, S. (1989). "Unified Formulation for Incompressible Flows," AIAA Paper 89-0122.
- Partom, I.S. (1987). "Application of the VOF Method to the Sloshing of a Fluid in a Partially Filled Cylindrical Container," *International Journal For Numerical Methods In Fluids*, Vol. 7, pp. 535-550.
- Patankar, S.V. (1980). *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp., New York.
- Ramshaw, J.D., and Trapp, J.A. (1976). "A Numerical Technique for Low-Speed Homogeneous Two-Phase Flow with Sharp Interfaces," *Journal of Computational Physics*, Vol. 21, pp. 438-453.

- Lord Rayleigh (1900). "Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density," in *Scientific Papers*, Vol. 2, University Press, Cambridge, pp. 200-207.
- Roe, P.L. (1981). "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, pp. 357-372.
- Roe, P.L. (1986). "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, pp. 337-365. Providence, RI.
- Roe, P.L. (1993). "Beyond the Riemann Problem, Part I," in *Algorithmic Trends in Computational Fluid Dynamics*, M.Y. Hussani, A. Kumar, and M.D. Salas (Eds.), Springer-Verlag, New York, pp. 341-367.
- Rogers, S.E., Kwak, D., and Kiris, C. (1989). "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems," AIAA-Paper 89-0463.
- Segel, L.A. (1977). *Mathematics Applied to Continuum Mechanics*, Macmillan, New York.
- Sicilian, J.M., and Hirt, C.W. (1984). "Numerical Simulation of Propellant Sloshing for Spacecraft," in *Forum on Unsteady Flow - Winter Annual Meeting of the ASME* P.H. Rothe (Ed.), New Orleans, LA, December 9-14, pp. 10-12.
- Sicilian, J.M., and Tegart, J.R. (1989). "Comparison of FLOW-3D Calculations with Very Large Amplitude Slosh Data," *Computational Experiments*, PVP-Vol. 176, presented at the 1989 ASME Pressure Vessels and Piping Conference, Honolulu, Hawaii, July 23-27, pp. 23-30.
- Simon, J. (1990). "Nonhomogeneous Viscous Incompressible Fluids: Existence of Velocity, Density, and Pressure," *SIAM Journal of Mathematical Analysis*, Vol. 21, No. 5, pp. 1093-1117.
- Spalding, D.B. (1974). "A method for computing steady and unsteady flows possessing discontinuities of density," Report 910/2, CHAM Ltd, 40 High Street, London SW19 5AU.
- Stoker, J.J. (1957). *Water Waves*, Interscience Publishers Inc., New York.

- Taylor, G.I. (1950). "The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I," *Proceedings of the Royal Society of London, Ser. A*, Vol. 201, pp. 192-196.
- Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W. (1985). *Numerical Grid Generation: Foundations and Applications*, North Holland, Amsterdam.
- Tryggvason, G. (1988). "Numerical Simulations of Rayleigh-Taylor Instability," *Journal of Computational Physics*, Vol. 75, pp. 253-282.
- Unverdi, S.O., and Tryggvason, G. (1992). "A Front Tracking Method for Viscous, Incompressible Multi-fluid Flows," *Journal of Computational Physics*, Vol. 100, pp.25-37.
- van Leer, B. (1985). "Upwind Difference Methods for Aerodynamic Problems Governed by the Euler Equations," *Lecture Notes in Mathematics*, Vol. 22, pp. 327-336.
- Veldman, A.E.P., and Vogels, M.E.S. (1984). "Axisymmetric Liquid Sloshing Under Low-Gravity Conditions," *Acta Astronautica*, Vol. 11, No. 10/11, pp. 641-649.
- Vinokur, M. (1989). "An Analysis of Finite-Difference and Finite-Volume Formulations of Conservation Laws," *Journal of Computational Physics*, Vol. 81, No. 1, pp. 1-52.
- Wang, S.S. and Stuhmiller, J.H. (1989). "Modified Partial-Cell Method For Free-Surface Incompressible Flow Simulations," *Numerical Heat Transfer*, Vol. 3, pp. 209-223.
- Walatka, P.P, Buning, P.G., Pierce, L., and Elson, P.A. (1990). *PLOT3D User's Manual - Version 3.6*, NASA Technical Memorandum 101067.
- Yakota, J.W., and Caughey, D.A. (1988). "LU Implicit Multigrid Algorithm for the Three-Dimensional Euler Equations," *AIAA Journal*, Vol. 26, No. 9, pp. 1061-1069.
- Yakota, J.W., (1990). "Diagonally Inverted Lower-Upper Factored Implicit Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," *AIAA Journal*, Vol. 28, No. 9, pp. 1642-1649.

- Yee, H.C., (1989). "A Class of High Resolution Explicit and Implicit Shock-Capturing Schemes," von Karman Institute for Fluid Dynamics, Lecture Series 1989-04, March 6 - 10, 1989.
- Yoon, S., Kwak, D., and Chang, L. (1989). "LU-SGS Implicit Algorithm for Three-Dimensional Incompressible Navier-Stokes Equations with Source Term," AIAA Paper 89-1964.
- Yoon, S., and Kwak, D. (1991). "An Implicit Three-Dimensional Navier-Stokes Solver for Compressible Flows," AIAA Paper 91-1555.
- Youngs, D.L. (1984). "Numerical Simulation of Turbulent Mixing by Rayleigh-Taylor Instability," *Physica 12D*, pp. 32-44.

APPENDIX A. SOME NOTES ON SURFACE TENSION

This appendix provides some additional details on the approximation of the surface tension force as discussed in Chapter 2. The ideas presented below draw primarily upon the works of Kataoka (1986) and Brackbill, Kothe, and Zemach (1992).

To begin, consider a continuous function $f(x, y, z, t)$ where $f > 0$ in the liquid, $f < 0$ in the gas, and $f = 0$ at the free surface. Based on this definition, one can introduce a characteristic function ϕ for each fluid as follows:

$$\phi_1(x, y, z, t) = H(f) \quad (\text{A.1})$$

$$\phi_2(x, y, z, t) = 1 - H(f) \quad (\text{A.2})$$

where $H(f)$ is the unit step function

$$H(f) = \begin{cases} 0 & f < 0 \\ 1 & f > 0 \end{cases} \quad (\text{A.3})$$

The characteristic function is also known as the local void fraction, as it takes on a value of one if the fluid occupies a given point in space and zero otherwise. It follows, then, that the fluid density field can be written in terms of the characteristic functions as

$$\rho = \rho_1 \phi_1 + \rho_2 \phi_2 \quad (\text{A.4})$$

It was shown in Chapter 2 that the local surface tension force $\vec{F}_s(x, y, z, t)$ could be expressed as

$$\vec{F}_s(x, y, z, t) = 2\sigma \kappa \hat{\nu} a_i \quad (\text{A.5})$$

where a_i is the local interfacial area concentration

$$a_i(x, y, z, t) = |\nabla f| \Delta(f) \quad (\text{A.6})$$

In the above, $\Delta(f)$ is the Dirac delta function, σ is the surface tension coefficient, κ is the surface curvature, and $\hat{\nu}$ is a unit vector on the free surface pointing in the direction of fluid 2, as shown in Fig. 2.2. Using the relation

$$\frac{d}{df} [H(f)] = \Delta(f) \quad (\text{A.7})$$

it can be shown that the local interfacial area concentration is related to the characteristic functions by

$$\nabla \phi_1 = -a_i \hat{\nu} \quad (\text{A.8})$$

$$\nabla \phi_2 = a_i \hat{\nu} \quad (\text{A.9})$$

Here, the gradient operator indicates differentiation in the sense of a distribution since the characteristic function is discontinuous. By the definition of density given by Eq. (A.4), it can be easily shown that

$$\nabla \rho = (\rho_1 - \rho_2) \nabla \phi_1 \quad (\text{A.10})$$

Therefore,

$$\frac{\nabla \rho}{\rho_1 - \rho_2} = -a_i \hat{v} \quad (\text{A.11})$$

Introducing this relation into the surface tension force yields

$$\vec{F}_s(x, y, z, t) = -\frac{2\sigma}{(\rho_1 - \rho_2)} \kappa \nabla \rho \quad (\text{A.12})$$

It is interesting to note that this formulation of the surface tension force is quite similar to that developed by Brackbill, Kothe, and Zemach (1992).

A finite volume discretization of the surface tension force can be developed in a straightforward way. For example, considering a finite control volume Ω , the integrated surface tension force can be approximated by

$$\int_{\Omega} \vec{F}_s d\Omega \approx -\frac{2\sigma\Omega}{\rho_1 - \rho_2} \bar{\kappa} \bar{\nabla \rho} \quad (\text{A.13})$$

where $\bar{\kappa}$ and $\bar{\nabla \rho}$ are suitable volume averages. Several methods for computing $\bar{\kappa}$ and $\bar{\nabla \rho}$ are described by Brackbill, Kothe, and Zemach (1992).

APPENDIX B. INVISCID FLUX JACOBIANS

Two Dimensional Form

The solution vector \mathbf{U} and the inviscid flux vector $\mathbf{F}_i \equiv \vec{\mathbf{F}}_i \cdot \vec{S}$ for the two dimensional two-fluid equations are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ P \end{bmatrix} \quad (B.1)$$

$$\mathbf{F}_i = \begin{bmatrix} \rho q \\ \rho qu + \beta PS_x \\ \rho qv + \beta PS_y \\ q \end{bmatrix} \quad (B.2)$$

where

$$q = \vec{V} \cdot \vec{S} = uS_x + vS_y \quad (B.3)$$

$$\vec{S} = S_x\hat{i} + S_y\hat{j} \quad (B.4)$$

The inviscid Jacobian matrix $[\mathbf{A}_i] \equiv \partial \mathbf{F}_i / \partial \mathbf{U}$ is given by

$$[\mathbf{A}_i] = \begin{bmatrix} 0 & S_x & S_y & 0 \\ -qu & q + uS_x & uS_y & \beta S_x \\ -qv & vS_x & q + vS_y & \beta S_y \\ -\frac{q}{\rho} & \frac{S_x}{\rho} & \frac{S_y}{\rho} & 0 \end{bmatrix} \quad (\text{B.5})$$

Three Dimensional Form

The solution vector \mathbf{U} and the inviscid flux vector $\mathbf{F}_i \equiv \vec{\mathbf{F}}_i \cdot \vec{S}$ for the three dimensional two-fluid equations are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ P \end{bmatrix} \quad (\text{B.6})$$

$$\mathbf{F}_i = \begin{bmatrix} \rho q \\ \rho qu + \beta PS_x \\ \rho qv + \beta PS_y \\ \rho qw + \beta PS_z \\ q \end{bmatrix} \quad (\text{B.7})$$

where

$$q = \vec{V} \cdot \vec{S} = uS_x + vS_y + wS_z \quad (\text{B.8})$$

$$\vec{S} = S_x \hat{i} + S_y \hat{j} + S_z \hat{k} \quad (\text{B.9})$$

The inviscid Jacobian matrix $[\mathbf{A}_i] \equiv \partial \mathbf{F}_i / \partial \mathbf{U}$ is given by

$$[\mathbf{A}_i] = \begin{bmatrix} 0 & S_x & S_y & S_z & 0 \\ -qu & q + uS_x & uS_y & uS_z & \beta S_x \\ -qv & vS_x & q + vS_y & vS_z & \beta S_y \\ -qw & wS_x & wS_y & q + wS_z & \beta S_z \\ -\frac{q}{\rho} & \frac{S_x}{\rho} & \frac{S_y}{\rho} & \frac{S_z}{\rho} & 0 \end{bmatrix} \quad (\text{B.10})$$

APPENDIX C. EIGENSYSTEM OF THE INVISCID FLUX JACOBIANS

The eigensystem associated with the inviscid flux Jacobians is developed in this appendix. As in Appendix A, the two and three dimensional forms are treated separately.

The m eigenvalues of the inviscid flux Jacobian matrix $[\mathbf{A}_i]$ ($m = 4$ for 2D, $m = 5$ for 3D) are derived from the determinant

$$\left| [\mathbf{A}_i] - \lambda_l [\mathbf{I}] \right| = 0 \quad (\text{C.1})$$

where λ_l denotes the l^{th} eigenvalue. The m right eigenvectors are then obtained by solving the following system for each eigenvalue:

$$([\mathbf{A}_i] - \lambda_l [\mathbf{I}]) \mathbf{V}_l = 0 \quad (\text{C.2})$$

where \mathbf{V}_l is the l^{th} right eigenvector of $[\mathbf{A}_i]$. The matrix whose columns are the right eigenvectors taken in order is denoted $[\mathbf{T}_i]$ and its inverse $[\mathbf{T}_i]^{-1}$. Since $[\mathbf{A}_i]$ has real eigenvalues and a linearly independant set of right eigenvectors (to be defined below), then the following similarity relation holds:

$$[\mathbf{A}_i] = [\mathbf{T}_i][\Lambda][\mathbf{T}_i]^{-1} \quad (\text{C.3})$$

Here, $[\Lambda]$ is the diagonal eigenvalue matrix

$$[\Lambda] = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \quad (\text{C.4})$$

Two Dimensional Form

For the two dimensional inviscid flux Jacobian presented in Appendix A, the eigenvalues are

$$\lambda_1 = \lambda_2 = q \quad (\text{C.5})$$

$$\lambda_{3,4} = \frac{1}{2}(q \pm c) \quad (\text{C.6})$$

where

$$c \equiv \sqrt{q^2 + 4\beta'(S_x^2 + S_y^2)} \quad (\text{C.7})$$

$$\beta' \equiv \frac{\beta}{\rho} \quad (\text{C.8})$$

and $\vec{S} = S_x \hat{i} + S_y \hat{j}$ is either \vec{S}^ϵ or \vec{S}^η .

The dependence of the right eigenvectors on the geometry produces minor differences in the elements of $[\mathbf{T}]$ and $[\mathbf{T}]^{-1}$ for each coordinate direction. Accordingly, the specific forms of these matrices for the \vec{S}^ϵ and \vec{S}^η cell faces are presented below. For convenience, the following notation is used:

$$S = \sqrt{S_x^2 + S_y^2} \quad (\text{C.9})$$

Also note that the metric derivatives required in these matrices ($\partial x/\partial\xi$, $\partial x/\partial\eta$, etc.) are approximated using finite differences.

- Cell face \vec{S}^ξ :

$$[\mathbf{T}_i^\xi] = \begin{bmatrix} 1 & 0 & \lambda_3 & \lambda_4 \\ u & \partial x/\partial\eta & u\lambda_3 + \beta' S_x & u\lambda_4 + \beta' S_x \\ v & \partial y/\partial\eta & v\lambda_3 + \beta' S_y & v\lambda_4 + \beta' S_y \\ 0 & 0 & -\lambda_4/\rho & -\lambda_3/\rho \end{bmatrix} \quad (\text{C.10})$$

$$[\mathbf{T}_i^\xi]^{-1} = \begin{bmatrix} 1 + q^2/\beta' S^2 & -qS_x/\beta' S^2 & -qS_y/\beta' S^2 & -\rho \\ (uS_y - vS_x)/S^2 & -S_y/S^2 & S_x/S^2 & 0 \\ -q\lambda_3/\beta' S^2 c & \lambda_3 S_x/\beta' S^2 c & \lambda_3 S_y/\beta' S^2 c & \rho/c \\ q\lambda_4/\beta' S^2 c & -\lambda_4 S_x/\beta' S^2 c & -\lambda_4 S_y/\beta' S^2 c & -\rho/c \end{bmatrix} \quad (\text{C.11})$$

where \vec{S} , q , λ_3 , λ_4 , and c are evaluated using \vec{S}^ξ .

- Cell face \vec{S}^η :

$$[\mathbf{T}_i^\eta] = \begin{bmatrix} 1 & 0 & \lambda_3 & \lambda_4 \\ u & \partial x/\partial\xi & u\lambda_3 + \beta' S_x & u\lambda_4 + \beta' S_x \\ v & \partial y/\partial\xi & v\lambda_3 + \beta' S_y & v\lambda_4 + \beta' S_y \\ 0 & 0 & -\lambda_4/\rho & -\lambda_3/\rho \end{bmatrix} \quad (\text{C.12})$$

$$[\mathbf{T}_i^\eta]^{-1} = \begin{bmatrix} 1 + q^2/\beta' S^2 & -qS_x/\beta' S^2 & -qS_y/\beta' S^2 & -\rho \\ -(uS_y - vS_x)/S^2 & -S_y/S^2 & S_x/S^2 & 0 \\ -q\lambda_3/\beta' S^2 c & \lambda_3 S_x/\beta' S^2 c & \lambda_3 S_y/\beta' S^2 c & \rho/c \\ q\lambda_4/\beta' S^2 c & -\lambda_4 S_x/\beta' S^2 c & -\lambda_4 S_y/\beta' S^2 c & -\rho/c \end{bmatrix} \quad (\text{C.13})$$

where \vec{S} , q , λ_3 , λ_4 , and c are evaluated using \vec{S}^η .

Three Dimensional Form

For the three dimensional inviscid flux Jacobian presented in Appendix A, the eigenvalues are

$$\lambda_1 = \lambda_2 = \lambda_3 = q \quad (\text{C.14})$$

$$\lambda_{4,5} = \frac{1}{2}(q \pm c) \quad (\text{C.15})$$

where

$$c \equiv \sqrt{q^2 + 4\beta'(S_x^2 + S_y^2 + S_z^2)} \quad (\text{C.16})$$

$$\beta' \equiv \frac{\beta}{\rho} \quad (\text{C.17})$$

and $\vec{S} = S_x \hat{i} + S_y \hat{j} + S_z \hat{k}$ is either \vec{S}^ϵ , \vec{S}^η , or \vec{S}^ζ .

The dependence of the right eigenvectors on the geometry produces minor differences in the elements of $[\mathbf{T}]$ and $[\mathbf{T}]^{-1}$ for each coordinate direction. Accordingly, the specific forms of these matrices for the \vec{S}^ϵ , \vec{S}^η , \vec{S}^ζ cell faces are presented below. For convenience, the following notation is used:

$$S = \sqrt{S_x^2 + S_y^2 + S_z^2} \quad (\text{C.18})$$

$$\vec{g}^\xi = \frac{\partial \vec{r}}{\partial \xi} \times \vec{S} \quad (\text{C.19})$$

$$\vec{g}^\eta = \frac{\partial \vec{r}}{\partial \eta} \times \vec{S} \quad (\text{C.20})$$

$$\vec{g}^\zeta = \frac{\partial \vec{r}}{\partial \zeta} \times \vec{S} \quad (\text{C.21})$$

Also note that the metric derivatives required in these matrices ($\partial x/\partial \xi$, $\partial x/\partial \eta$, etc.) are approximated using finite differences.

- Cell face \vec{S}^ξ :

$$[\mathbf{T}_i^\xi] = \begin{bmatrix} 1 & 0 & 0 & \lambda_4 & \lambda_5 \\ u & \partial x / \partial \eta & \partial x / \partial \zeta & u\lambda_4 + \beta' S_x & u\lambda_5 + \beta' S_x \\ v & \partial y / \partial \eta & \partial y / \partial \zeta & v\lambda_4 + \beta' S_y & v\lambda_5 + \beta' S_y \\ w & \partial z / \partial \eta & \partial z / \partial \zeta & w\lambda_4 + \beta' S_z & w\lambda_5 + \beta' S_z \\ 0 & 0 & 0 & -\lambda_5 / \rho & -\lambda_4 / \rho \end{bmatrix} \quad (\text{C.22})$$

$$[\mathbf{T}_i^\xi]^{-1} = \begin{bmatrix} 1 + q^2 / \beta' S^2 & -qS_x / \beta' S^2 & -qS_y / \beta' S^2 & -qS_z / \beta' S^2 & -\rho \\ -\vec{g}^\xi \cdot \vec{V} / S^2 & \vec{g}^\xi \cdot \hat{i} / S^2 & \vec{g}^\xi \cdot \hat{j} / S^2 & \vec{g}^\xi \cdot \hat{k} / S^2 & 0 \\ \vec{g}^\eta \cdot \vec{V} / S^2 & -\vec{g}^\eta \cdot \hat{i} / S^2 & -\vec{g}^\eta \cdot \hat{j} / S^2 & -\vec{g}^\eta \cdot \hat{k} / S^2 & 0 \\ -q\lambda_4 / \beta' S^2 c & \lambda_4 S_x / \beta' S^2 c & \lambda_4 S_y / \beta' S^2 c & \lambda_4 S_z / \beta' S^2 c & \rho / c \\ q\lambda_5 / \beta' S^2 c & -\lambda_5 S_x / \beta' S^2 c & -\lambda_5 S_y / \beta' S^2 c & -\lambda_5 S_z / \beta' S^2 c & -\rho / c \end{bmatrix} \quad (\text{C.23})$$

where \vec{S} , q , λ_4 , λ_5 , and c are evaluated using \vec{S}^ξ .

- Cell face \vec{S}^η :

$$[\mathbf{T}_i^\eta] = \begin{bmatrix} 1 & 0 & 0 & \lambda_4 & \lambda_5 \\ u & \partial x / \partial \zeta & \partial x / \partial \xi & u\lambda_4 + \beta' S_x & u\lambda_5 + \beta' S_x \\ v & \partial y / \partial \zeta & \partial y / \partial \xi & v\lambda_4 + \beta' S_y & v\lambda_5 + \beta' S_y \\ w & \partial z / \partial \zeta & \partial z / \partial \xi & w\lambda_4 + \beta' S_z & w\lambda_5 + \beta' S_z \\ 0 & 0 & 0 & -\lambda_5 / \rho & -\lambda_4 / \rho \end{bmatrix} \quad (\text{C.24})$$

$$[\mathbf{T}_i^\eta]^{-1} = \begin{bmatrix} 1 + q^2 / \beta' S^2 & -qS_x / \beta' S^2 & -qS_y / \beta' S^2 & -qS_z / \beta' S^2 & -\rho \\ -\vec{g}^\xi \cdot \vec{V} / S^2 & \vec{g}^\xi \cdot \hat{i} / S^2 & \vec{g}^\xi \cdot \hat{j} / S^2 & \vec{g}^\xi \cdot \hat{k} / S^2 & 0 \\ \vec{g}^\xi \cdot \vec{V} / S^2 & -\vec{g}^\xi \cdot \hat{i} / S^2 & -\vec{g}^\xi \cdot \hat{j} / S^2 & -\vec{g}^\xi \cdot \hat{k} / S^2 & 0 \\ -q\lambda_4 / \beta' S^2 c & \lambda_4 S_x / \beta' S^2 c & \lambda_4 S_y / \beta' S^2 c & \lambda_4 S_z / \beta' S^2 c & \rho / c \\ q\lambda_5 / \beta' S^2 c & -\lambda_5 S_x / \beta' S^2 c & -\lambda_5 S_y / \beta' S^2 c & -\lambda_5 S_z / \beta' S^2 c & -\rho / c \end{bmatrix} \quad (\text{C.25})$$

where \vec{S} , q , λ_4 , λ_5 , and c are evaluated using \vec{S}^η .

- Cell face \vec{S}^ξ :

$$[\mathbf{T}_i^\xi] = \begin{bmatrix} 1 & 0 & 0 & \lambda_4 & \lambda_5 \\ u & \partial x / \partial \xi & \partial x / \partial \eta & u\lambda_4 + \beta' S_x & u\lambda_5 + \beta' S_x \\ v & \partial y / \partial \xi & \partial y / \partial \eta & v\lambda_4 + \beta' S_y & v\lambda_5 + \beta' S_y \\ w & \partial z / \partial \xi & \partial z / \partial \eta & w\lambda_4 + \beta' S_z & w\lambda_5 + \beta' S_z \\ 0 & 0 & 0 & -\lambda_5 / \rho & -\lambda_4 / \rho \end{bmatrix} \quad (\text{C.26})$$

$$[\mathbf{T}_i^\zeta]^{-1} = \begin{bmatrix} 1 + q^2/\beta' S^2 & -qS_x/\beta' S^2 & -qS_y/\beta' S^2 & -qS_z/\beta' S^2 & -\rho \\ -\vec{g}^\eta \cdot \vec{V}/S^2 & \vec{g}^\eta \cdot \hat{i}/S^2 & \vec{g}^\eta \cdot \hat{j}/S^2 & \vec{g}^\eta \cdot \hat{k}/S^2 & 0 \\ \vec{g}^\xi \cdot \vec{V}/S^2 & -\vec{g}^\xi \cdot \hat{i}/S^2 & -\vec{g}^\xi \cdot \hat{j}/S^2 & -\vec{g}^\xi \cdot \hat{k}/S^2 & 0 \\ -q\lambda_4/\beta' S^2 c & \lambda_4 S_x/\beta' S^2 c & \lambda_4 S_y/\beta' S^2 c & \lambda_4 S_z/\beta' S^2 c & \rho/c \\ q\lambda_5/\beta' S^2 c & -\lambda_5 S_x/\beta' S^2 c & -\lambda_5 S_y/\beta' S^2 c & -\lambda_5 S_z/\beta' S^2 c & -\rho/c \end{bmatrix} \quad (\text{C.27})$$

where \vec{S} , q , λ_4 , λ_5 , and c are evaluated using \vec{S}^ζ .

**APPENDIX D. APPROXIMATE LINEARIZATION OF THE
VISCOSUS FLUX**

As described in Section 4.5, the viscous flux is linearized as

$$\mathbf{F}_v^{m+1} = \mathbf{F}_v^m + \delta\mathbf{F}_v \quad (\text{D.1})$$

where

$$\mathbf{F}_v \equiv \vec{\mathbf{F}}_v \cdot \vec{S}_{i+\frac{1}{2}}^\xi \quad (\text{D.2})$$

$$\delta\mathbf{F}_v \equiv \left(\frac{\partial \mathbf{F}_v}{\partial \tau} \right)^m \Delta\tau \quad (\text{D.3})$$

Assuming that \mathbf{F}_v is a function of \mathbf{U} and the Cartesian derivatives of \mathbf{U} , then Eq. (D.3) can be expanded as

$$\delta\mathbf{F}_v = \left[\left(\frac{\partial \mathbf{F}_v}{\partial \mathbf{U}} \right)^m \frac{\partial \mathbf{U}}{\partial \tau} + \left(\frac{\partial \mathbf{F}_v}{\partial \mathbf{U}_x} \right)^m \frac{\partial \mathbf{U}_x}{\partial \tau} + \left(\frac{\partial \mathbf{F}_v}{\partial \mathbf{U}_y} \right)^m \frac{\partial \mathbf{U}_y}{\partial \tau} + \left(\frac{\partial \mathbf{F}_v}{\partial \mathbf{U}_z} \right)^m \frac{\partial \mathbf{U}_z}{\partial \tau} \right] \Delta\tau \quad (\text{D.4})$$

where the subscripts x , y , and z denote differentiation with respect to the given Cartesian coordinate. Note that the derivatives of \mathbf{F}_v represent 4×4 matrices in the two-dimensional case and 5×5 matrices in the three dimensional case.

A simplification is now introduced wherein $\partial\mathbf{F}_v/\partial\mathbf{U}$ is assumed to have a smaller influence on $\delta\mathbf{F}_v$ than the other Jacobians, and hence can be neglected. In the remaining terms, the order of time and spatial differentiation can be interchanged to give

$$\delta\mathbf{F}_v \approx \left[+ \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_x} \right)^m \frac{\partial\mathbf{U}_\tau}{\partial x} + \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_y} \right)^m \frac{\partial\mathbf{U}_\tau}{\partial y} + \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_z} \right)^m \frac{\partial\mathbf{U}_\tau}{\partial z} \right] \Delta\tau \quad (\text{D.5})$$

where $\mathbf{U}_\tau = \partial\mathbf{U}/\partial\tau$. Noting that

$$\delta\mathbf{U} \approx \mathbf{U}_\tau \Delta\tau \quad (\text{D.6})$$

then Eq. (D.5) becomes

$$\delta\mathbf{F}_v \approx \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_x} \right)^m \frac{\partial}{\partial x} (\delta\mathbf{U}) + \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_y} \right)^m \frac{\partial}{\partial y} (\delta\mathbf{U}) + \left(\frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_z} \right)^m \frac{\partial}{\partial z} (\delta\mathbf{U}) \quad (\text{D.7})$$

The above can be rewritten in a more compact form as follows:

$$\delta\mathbf{F}_v \approx [\vec{\mathbf{A}}_v] \cdot \nabla (\delta\mathbf{U}) \quad (\text{D.8})$$

where

$$[\vec{\mathbf{A}}_v] \equiv [\mathbf{A}_{vx}] \hat{i} + [\mathbf{A}_{vy}] \hat{j} + [\mathbf{A}_{vz}] \hat{k} \quad (\text{D.9})$$

$$[\mathbf{A}_{vx}] \equiv \frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_x} \quad (\text{D.10})$$

$$[\mathbf{A}_{vy}] \equiv \frac{\partial\mathbf{F}_v}{\partial\mathbf{U}_y} \quad (\text{D.11})$$

$$[\mathbf{A}_{vz}] \equiv \frac{\partial \mathbf{F}_v}{\partial \mathbf{U}_z} \quad (\text{D.12})$$

For a given cell face, the Jacobians $[\mathbf{A}_{vx}]$, $[\mathbf{A}_{vy}]$, and $[\mathbf{A}_{vz}]$ can be computed using interpolated data, and the gradient term can be approximated according to the approach described in Section 3.4.6. Unfortunately, the discretized form of the gradient leads to a dependency of the unknowns at a given interior grid point on 8 neighbor points in two dimensions and 18 neighbor points in three dimensions. It is therefore common to discard non-orthogonal terms in the gradient operator (Rogers et al., 1989). Such an approximation will be derived for the $i + \frac{1}{2}$ cell face below.

Using the non-conservative form of the gradient operator at the $i + \frac{1}{2}$ face and discarding the η and ζ derivative terms gives

$$\delta \mathbf{F}_v \approx [\vec{\mathbf{A}}_v] \cdot \nabla \xi \frac{\partial}{\partial \xi} (\delta \mathbf{U}) \quad (\text{D.13})$$

To simplify the numerical calculations, the metric term $\nabla \xi$ is approximated as

$$(\nabla \xi)_{i+\frac{1}{2}} \approx \left(\frac{\vec{S}^\xi \Delta \xi}{\vec{S}^\xi \cdot \Delta l^\xi} \right)_{i+\frac{1}{2}} \quad (\text{D.14})$$

Employing a second order finite difference approximation for the ξ derivative

$$\frac{\partial}{\partial \xi} (\delta \mathbf{U}) \approx \frac{\delta \mathbf{U}_{i+1} - \delta \mathbf{U}_i}{\Delta \xi} \quad (\text{D.15})$$

and defining

$$[\mathbf{D}_v]_{i+\frac{1}{2}} \equiv \left([\vec{\mathbf{A}}_v] \cdot \frac{\nabla \xi}{\Delta \xi} \right)_{i+\frac{1}{2}} \quad (\text{D.16})$$

Eq. (D.13) can be written

$$\delta \mathbf{F}_v \approx [\mathbf{D}_v]_{i+\frac{1}{2}} (\delta \mathbf{U}_{i+1} - \delta \mathbf{U}) \quad (\text{D.17})$$

Similar approximate linearizations can be derived for cell faces $j + \frac{1}{2}$ and $k + \frac{1}{2}$, namely

$$\delta \mathbf{F}_v \approx [\mathbf{D}_v]_{j+\frac{1}{2}} (\delta \mathbf{U}_{j+1} - \delta \mathbf{U}) \quad (\text{D.18})$$

$$\delta \mathbf{F}_v \approx [\mathbf{D}_v]_{k+\frac{1}{2}} (\delta \mathbf{U}_{k+1} - \delta \mathbf{U}) \quad (\text{D.19})$$

where

$$[\mathbf{D}_v]_{j+\frac{1}{2}} \equiv ([\vec{\mathbf{A}}_v] \cdot \nabla \eta)_{j+\frac{1}{2}} \quad (\text{D.20})$$

$$[\mathbf{D}_v]_{k+\frac{1}{2}} \equiv ([\vec{\mathbf{A}}_v] \cdot \nabla \zeta)_{k+\frac{1}{2}} \quad (\text{D.21})$$

The matrix $[\mathbf{D}_v]_{i+\frac{1}{2}}$ can be easily derived from the definition of the viscous flux vector and the approximations given above. The resulting two and three dimensional forms are given below.

- Two-Dimensional Form

$$[\mathbf{D}_v]_{i+\frac{1}{2}} = \left(\frac{\mu}{\rho \vec{S}^\xi \cdot \Delta l^\xi} \right)_{i+\frac{1}{2}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (S_x^\xi)^2 + |\vec{S}^\xi|^2 & S_x^\xi S_y^\xi & 0 \\ 0 & S_x^\xi S_y^\xi & (S_y^\xi)^2 + |\vec{S}^\xi|^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{D.22})$$

- Three-Dimensional Form

$$[\mathbf{D}_v]_{i+\frac{1}{2}} = \left(\frac{\mu}{\rho \vec{S}^\xi \cdot \Delta l^\xi} \right)_{i+\frac{1}{2}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & (S_x^\xi)^2 + |\vec{S}^\xi|^2 & S_x^\xi S_y^\xi & S_x^\xi S_z^\xi & 0 \\ 0 & S_x^\xi S_y^\xi & (S_y^\xi)^2 + |\vec{S}^\xi|^2 & S_y^\xi S_z^\xi & 0 \\ 0 & S_x^\xi S_z^\xi & S_y^\xi S_z^\xi & (S_z^\xi)^2 + |\vec{S}^\xi|^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{D.23})$$

The matrices $[\mathbf{D}_v]_{j+\frac{1}{2}}$, and $[\mathbf{D}_v]_{k+\frac{1}{2}}$ are constructed in an analogous fashion using the appropriate geometric terms.

APPENDIX E. BODY FORCE LINEARIZATION

If surface tension forces are neglected, the source term for the momentum equation can be written in column vector form as

$$\mathbf{B} = \begin{bmatrix} 0 \\ B_x \\ B_y \\ B_z \end{bmatrix} \quad (\text{E.1})$$

where

$$B_x = \rho \left[-g_{ex} + \ddot{R}_x + 2(\omega_y w - \omega_z v) + \dot{\omega}_y z - \dot{\omega}_z y + \omega_y (\omega_x y - \omega_y x) - \omega_z (\omega_z x - \omega_x z) \right] \quad (\text{E.2})$$

$$B_y = \rho \left[-g_{ey} + \ddot{R}_y + 2(\omega_z u - \omega_x w) + \dot{\omega}_z x - \dot{\omega}_x z + \omega_z (\omega_y z - \omega_z y) - \omega_x (\omega_x y - \omega_y x) \right] \quad (\text{E.3})$$

$$B_z = \rho \left[-g_{ez} + \ddot{R}_z + 2(\omega_x v - \omega_y u) + \dot{\omega}_x y - \dot{\omega}_y x + \omega_x (\omega_z x - \omega_x z) - \omega_y (\omega_y z - \omega_z y) \right] \quad (\text{E.4})$$

$$\vec{g}_e = g_{ex}\hat{i} + g_{ey}\hat{j} + g_{ez}\hat{k} \quad (\text{E.5})$$

$$\vec{\omega} = \omega_x\hat{i} + \omega_y\hat{j} + \omega_z\hat{k} \quad (\text{E.6})$$

$$\frac{d\vec{\omega}}{dt} = \dot{\omega}_x\hat{i} + \dot{\omega}_y\hat{j} + \dot{\omega}_z\hat{k} \quad (\text{E.7})$$

$$\frac{d^2\vec{R}}{dt^2} = \ddot{R}_x\hat{i} + \ddot{R}_y\hat{j} + \ddot{R}_z\hat{k} \quad (\text{E.8})$$

Noting that \vec{g}_e , $\vec{\omega}$, $d\vec{\omega}/dt$, and $d^2\vec{\omega}/dt^2$ are known functions for time, the Jacobian matrix of \mathbf{B} can be easily determined. The result is

$$\frac{\partial \mathbf{B}}{\partial \mathbf{U}} \equiv [\mathbf{G}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ B_x/\rho & 0 & -2\omega_x & 2\omega_y & 0 \\ B_y/\rho & 2\omega_z & 0 & -2\omega_x & 0 \\ B_z/\rho & -2\omega_y & 2\omega_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{E.9})$$

APPENDIX F. BLOCK COEFFICIENTS AND RIGHT HAND SIDE VECTOR

The discrete, linearized system of equations was presented in Chapter 5 as

$$\begin{aligned} \frac{\Omega}{\Delta\tau} ([\mathbf{I}] + [\mathbf{H}]) \delta \mathbf{U} + [\mathbf{C}_P] \delta \mathbf{U} + [\mathbf{C}_E] \delta \mathbf{U}_{i+1} + [\mathbf{C}_W] \delta \mathbf{U}_{i-1} + [\mathbf{C}_N] \delta \mathbf{U}_{j+1} \\ + [\mathbf{C}_S] \delta \mathbf{U}_{j-1} + [\mathbf{C}_F] \delta \mathbf{U}_{k+1} + [\mathbf{C}_B] \delta \mathbf{U}_{k-1} = -\Delta\tau \mathbf{R}' \end{aligned} \quad (\text{F.1})$$

The block coefficients and right hand side vector for Eq. (F.1) are defined below:

$$[\mathbf{C}_P] = [\mathbf{C}_{Pe}] + [\mathbf{C}_{Pw}] + [\mathbf{C}_{Pn}] + [\mathbf{C}_{Ps}] + [\mathbf{C}_{Pf}] + [\mathbf{C}_{Pb}] \quad (\text{F.2})$$

$$[\mathbf{C}_{Pe}] = \begin{cases} [\mathbf{I}_\theta][\mathbf{A}_i]_e[\mathbf{C}_{e1}^b] & i = imax - 1 \\ [\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{i+\frac{1}{2}} & 2 \leq i \leq imax - 2 \end{cases} \quad (\text{F.3})$$

$$[\mathbf{C}_{Pw}] = \begin{cases} -[\mathbf{I}_\theta][\mathbf{A}_i]_w[\mathbf{C}_{w1}^b] & i = 2 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{i-\frac{1}{2}} & 3 \leq i \leq imax - 1 \end{cases} \quad (\text{F.4})$$

$$[\mathbf{C}_{Pn}] = \begin{cases} [\mathbf{I}_\theta][\mathbf{A}_i]_n[\mathbf{C}_{n1}^b] & j = jmax - 1 \\ [\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{j+\frac{1}{2}} & 2 \leq j \leq jmax - 2 \end{cases} \quad (\text{F.5})$$

$$[\mathbf{C}_{Ps}] = \begin{cases} -[\mathbf{I}_\theta][\mathbf{A}_i]_s[\mathbf{C}_{s1}^b] & j = 2 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{j-\frac{1}{2}} & 3 \leq j \leq jmax - 1 \end{cases} \quad (\text{F.6})$$

$$[\mathbf{C}_{Pf}] = \begin{cases} [\mathbf{I}_\theta][\mathbf{A}_i]_f[\mathbf{C}_{f1}^b] & k = kmax - 1 \\ [\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{k+\frac{1}{2}} & 2 \leq k \leq kmax - 2 \end{cases} \quad (\text{F.7})$$

$$[\mathbf{C}_{Pb}] = \begin{cases} -[\mathbf{I}_\theta][\mathbf{A}_i]_b[\mathbf{C}_{b1}^b] & k = 2 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{k-\frac{1}{2}} & 3 \leq k \leq kmax - 1 \end{cases} \quad (\text{F.8})$$

$$[\mathbf{C}_E] = \begin{cases} [\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{i+\frac{1}{2}} - [\mathbf{A}_i]_w[\mathbf{C}_{w2}^b] & i = 2 \\ 0 & i = imax - 1 \\ [\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{i+\frac{1}{2}} & 3 \leq i \leq imax - 2 \end{cases} \quad (\text{F.9})$$

$$[\mathbf{C}_W] = \begin{cases} 0 & i = 2 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{i-\frac{1}{2}} + [\mathbf{A}_i]_e[\mathbf{C}_{e2}^b] & i = imax - 1 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{i-\frac{1}{2}} & 3 \leq i \leq imax - 2 \end{cases} \quad (\text{F.10})$$

$$[\mathbf{C}_N] = \begin{cases} [\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{j+\frac{1}{2}} - [\mathbf{A}_i]_s[\mathbf{C}_{s2}^b] & j = 2 \\ 0 & j = jmax - 1 \\ [\mathbf{I}_\theta]([\mathbf{A}_i^-] - [\mathbf{D}_v])_{j+\frac{1}{2}} & 3 \leq j \leq jmax - 2 \end{cases} \quad (\text{F.11})$$

$$[\mathbf{C}_S] = \begin{cases} 0 & j = 2 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{j-\frac{1}{2}} + [\mathbf{A}_i]_n[\mathbf{C}_{n2}^b] & j = jmax - 1 \\ -[\mathbf{I}_\theta]([\mathbf{A}_i^+] + [\mathbf{D}_v])_{j-\frac{1}{2}} & 3 \leq j \leq jmax - 2 \end{cases} \quad (\text{F.12})$$

$$[\mathbf{C}_F] = \begin{cases} [\mathbf{I}_\theta] ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{k+\frac{1}{2}} - [\mathbf{A}_i]_b [\mathbf{C}_{b2}^b] & k = 2 \\ 0 & k = kmax - 1 \\ [\mathbf{I}_\theta] ([\mathbf{A}_i^-] - [\mathbf{D}_v])_{k+\frac{1}{2}} & 3 \leq k \leq kmax - 2 \end{cases} \quad (\text{F.13})$$

$$[\mathbf{C}_B] = \begin{cases} 0 & k = 2 \\ -[\mathbf{I}_\theta] ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{k-\frac{1}{2}} + [\mathbf{A}_i]_f [\mathbf{C}_{f2}^b] & k = kmax - 1 \\ -[\mathbf{I}_\theta] ([\mathbf{A}_i^+] + [\mathbf{D}_v])_{k-\frac{1}{2}} & 3 \leq k \leq kmax - 2 \end{cases} \quad (\text{F.14})$$

and the residual is redefined as

$$\mathbf{R}' = \mathbf{R} + \mathbf{CB}_E + \mathbf{CB}_W + \mathbf{CB}_N + \mathbf{CB}_S + \mathbf{CB}_F + \mathbf{CB}_B \quad (\text{F.15})$$

$$\mathbf{CB}_E = \begin{cases} 0 & 2 \leq i \leq imax - 2 \\ [\mathbf{I}_\theta] [\mathbf{A}_i]_e \mathbf{C}_{e3}^b & i = imax - 1 \end{cases} \quad (\text{F.16})$$

$$\mathbf{CB}_W = \begin{cases} 0 & 3 \leq i \leq imax - 1 \\ -[\mathbf{I}_\theta] [\mathbf{A}_i]_w \mathbf{C}_{w3}^b & i = 2 \end{cases} \quad (\text{F.17})$$

$$\mathbf{CB}_N = \begin{cases} 0 & 2 \leq j \leq jmax - 2 \\ [\mathbf{I}_\theta] [\mathbf{A}_i]_n \mathbf{C}_{n3}^b & j = jmax - 1 \end{cases} \quad (\text{F.18})$$

$$\mathbf{CB}_S = \begin{cases} 0 & 3 \leq j \leq jmax - 1 \\ -[\mathbf{I}_\theta] [\mathbf{A}_i]_s \mathbf{C}_{s3}^b & j = 2 \end{cases} \quad (\text{F.19})$$

$$\mathbf{CB}_F = \begin{cases} 0 & 2 \leq k \leq kmax - 2 \\ [\mathbf{I}_\theta] [\mathbf{A}_i]_f \mathbf{C}_{f3}^b & k = kmax - 2 \end{cases} \quad (\text{F.20})$$

$$\mathbf{CB}_B = \begin{cases} 0 & 3 \leq k \leq kmax - 1 \\ -[\mathbf{L}_\theta][\mathbf{A}_i]_b \mathbf{C}_{b3}^b & k = 2 \end{cases} \quad (\text{F.21})$$

**APPENDIX G. CHANDRASEHKAR'S ANALYSIS OF THE
VISCOUS RAYLEIGH-TAYLOR INSTABILITY**

In his linear perturbation analysis of the viscous Rayleigh-Taylor problem, Chandrasehkar (1961) established the following perturbation growth rate relation for the case of uniform kinematic viscosity:

$$Q = \frac{y - 1}{\alpha_1 - \alpha_2} [y^3 + (1 + 4\alpha_1\alpha_2)y^2 + (3 + 8\alpha_1\alpha_2)y - (1 - 4\alpha_1\alpha_2)] \quad (G.1)$$

In the above,

$$Q = \frac{g_e}{k^3 \nu^2} \quad (G.2)$$

$$k = \frac{2\pi}{\lambda_p} \quad (G.3)$$

$$\alpha_1 = \rho_1 / (\rho_1 + \rho_2) \quad (G.4)$$

$$\alpha_2 = \rho_2 / (\rho_1 + \rho_2) \quad (G.5)$$

$$y = \sqrt{1 + \frac{n}{k^2 \nu}} \quad (G.6)$$

and the subscripts 1 and 2 denote the heavy and light fluids, respectively.

For the density ratio considered in Chapter 7, Eq. (G.1) becomes

$$Q = \frac{1}{3}(y - 1)(9y^3 + 17y^2 + 8y - 1) \quad (\text{G.7})$$

This equation can be used to establish the relationship between the perturbation wavelength and the growth rate using the following procedure (ν is assumed to be known):

1. Specify a value for the perturbation wavelength, λ_p .
2. Calculate Q from Eqs. (G.2) and (G.3).
3. Solve Eq. (G.7) for y .
4. Derive the growth rate from Eq. (G.6).

APPENDIX H. STEADY STATE FREE SURFACE SHAPES FOR SPHERICAL CONTAINERS

This appendix describes a general method for computing steady state free surface shapes for partially-filled spherical containers undergoing a simple orbital motion. This method is applicable to the axisymmetric case by setting the orbital radius h to zero.

Derivation of the Free Surface Equation

Consider a partially-filled spherical container of radius R , which rotates at a constant angular velocity $\vec{\omega}_z$ about the Z axis of an inertial frame (X, Y, Z) , as shown in Fig. H.1. A noninertial frame (x, y, z) is fixed to the sphere with its origin at the center of the sphere. The path of the center of the sphere is constrained to a circle of radius h in the $X - Y$ plane, with the vertical axis z parallel to the axis of rotation.

At steady state, the vector momentum equation for an incompressible fluid referred to the noninertial frame reduces to

$$\frac{1}{\rho} \nabla p = \frac{d^2 \vec{R}}{dt^2} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) - \vec{g}_e \quad (\text{H.1})$$

The vectors quantities $d^2 \vec{R}/dt^2$, $\vec{\omega}$, and \vec{g}_e can be written in terms of unit vectors \hat{i} ,

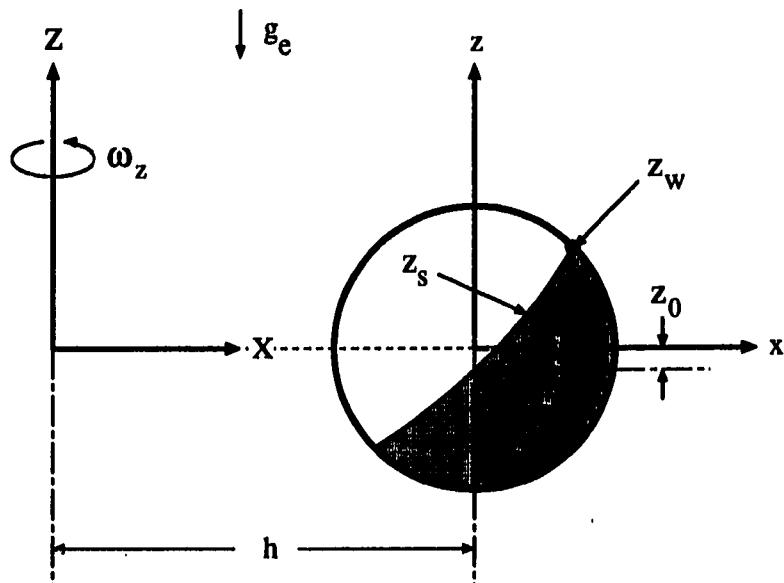


Figure H.1: Illustration of rotating spherical tank

\hat{i} , \hat{k} associated with the noninertial frame as follows:

$$\frac{d^2 \vec{R}}{dt^2} = -h\omega_z^2 \hat{i} \quad (\text{H.2})$$

$$\vec{\omega} = \omega_z \hat{k} \quad (\text{H.3})$$

$$\vec{g}_e = g_e \hat{k} \quad (\text{H.4})$$

Introducing these expressions into Eq. (H.1) and expanding into vector components gives

$$\frac{1}{\rho} \left(\frac{\partial p}{\partial x} \hat{i} + \frac{\partial p}{\partial y} \hat{j} + \frac{\partial p}{\partial z} \hat{k} \right) = \omega_z^2 (h + x) \hat{i} + \omega_z^2 y \hat{j} - g_e \hat{k} \quad (\text{H.5})$$

It is assumed that the pressure field can be expressed as the sum of three functions

$$p(x, y, z) = p_1(x) + p_2(y) + p_3(z) \quad (\text{H.6})$$

Substituting this form into Eq. (H.5) and considering each component in turn leads to the following differential equations:

$$\frac{1}{\rho} \frac{dp}{dx} = \omega_z^2(h + x) \quad (\text{H.7})$$

$$\frac{1}{\rho} \frac{dp}{dy} = \omega_z^2 y \quad (\text{H.8})$$

$$\frac{1}{\rho} \frac{dp}{dz} = -g_e \quad (\text{H.9})$$

Solving Eqs. (H.7) – (H.9) for functions p_1 , p_2 , and p_3 gives

$$p_1(x) = \frac{1}{2} \rho \omega_z^2 (h + x)^2 + C_1 \quad (\text{H.10})$$

$$p_2(y) = \frac{1}{2} \rho \omega_z^2 y^2 + C_2 \quad (\text{H.11})$$

$$p_3(z) = -\rho g_e z + C_3 \quad (\text{H.12})$$

where C_1 , C_2 , and C_3 are constants. Adding these solutions and combining the constants yields

$$p(x, y, z) = \frac{1}{2} \rho \omega_z^2 [(h + x)^2 + y^2] - \rho g_e z + C \quad (\text{H.13})$$

If the pressure at the free surface is assumed to be uniform, and the intersection of the free surface with the z axis is defined as z_0 , the equation for the free surface can be derived from Eq. (H.13) as

$$z = z_0 + \alpha(2hx + x^2 + y^2) \quad (\text{H.14})$$

where $\alpha = \omega_z^2/2g_e$. To simplify the solution procedure described below, this equation is transformed to cylindrical coordinates (r, θ, z) , yielding

$$z(r, \theta) = z_0 + \alpha(r^2 + 2hr \cos \theta) \quad (\text{H.15})$$

For a given fill ratio, a unique value of the constant z_0 can be determined numerically. This is the subject of the next section.

Numerical Solution Procedure

Consider the intersection of free surface equation, $z(r, \theta)$, with the walls of the spherical container. It can be seen that, in general, the intersection point (r_w, z_w) is a function of θ . Furthermore, depending on the fill ratio, $z_w(\theta)$ will satisfy one of the following conditions:

- $z_w(\theta) \geq 0$ for all θ
- $z_w(\theta) \geq 0$ for $0 \leq \theta \leq \theta_c$ and $z_w(\theta) < 0$ for $\theta_c < \theta \leq \pi$.
- $z_w(\theta) \leq 0$ for all θ

For the second case, it follows that $z_w = 0$ and $r_w = R$ at θ_c , which, from Eq. (H.15), leads to the following relation:

$$\theta_c = \cos^{-1} \left[\frac{-(R^2 + z_0/\alpha)}{2hR} \right] \quad (\text{H.16})$$

Now, let V_s denote the volume of the sphere, and V_f the volume of the fluid. By symmetry, half the difference in these volumes can be computed by integration as follows:

$$\begin{aligned} \frac{1}{2}(V_s - V_f) &= \int_0^{\theta_c} \int_0^{r_w} [\sqrt{R^2 - r^2} - z(r, \theta)] r dr d\theta \\ &+ \int_{\theta_c}^{\pi} \left\{ \int_0^R 2\sqrt{R^2 - r^2} r dr - \int_0^{r_w} [z(r, \theta) + \sqrt{R^2 - r^2}] r dr \right\} d\theta \end{aligned} \quad (\text{H.17})$$

Since $r_w = r_w(\theta)$, the integration in θ must be computed numerically. The integration in r , however, can be carried out analytically as follows:

$$\begin{aligned} I_1 &\equiv \int_0^{r_w} [\sqrt{R^2 - r^2} - z(r, \theta)] r dr \\ &= \int_0^{r_w} [r\sqrt{R^2 - r^2} - z_0 - \alpha(r^2 + 2hr \cos \theta)] dr \\ &= \frac{1}{3}R^3 - \frac{1}{3}(R^2 - r_w^2)^{\frac{3}{2}} - \frac{1}{2}z_0 r_w^2 - \frac{2}{3}\alpha h \cos \theta r_w^3 - \frac{1}{4}\alpha r_w^4 \end{aligned}$$

$$I_2 \equiv \int_0^R 2\sqrt{R^2 - r^2} r dr = \frac{2}{3}R^3$$

$$I_3 \equiv \int_0^{r_w} [z(r, \theta) + \sqrt{R^2 - r^2}] r dr$$

$$\begin{aligned}
&= \int_0^{r_w} [r\sqrt{R^2 - r^2} + z_0 + \alpha(r^2 + 2hr \cos \theta)] dr \\
&= \frac{1}{3}R^3 - \frac{1}{3}(R^2 - r_w^2)^{\frac{3}{2}} + \frac{1}{2}z_0r_w^2 + \frac{2}{3}\alpha h \cos \theta r_w^3 + \frac{1}{4}\alpha r_w^4
\end{aligned}$$

Introducing these integrals into Eq. (H.17) and rearranging gives

$$\frac{1}{2}(V_s - V_f) - \int_0^{\theta_c} I_1 d\theta - \int_{\theta_c}^{\pi} (I_2 - I_3) d\theta \equiv \mathcal{F}(r_w, z_0) = 0 \quad (\text{H.18})$$

A numerical procedure can now be defined for computing z_0 for given values of V_f , R , and h :

1. Guess z_0 .
2. Based on the guessed z_0 , determine θ_c as follows:
 - If $z_w(\theta) \geq 0$ for all θ , then set $\theta_c = \pi$.
 - If $z_w(\theta) \leq 0$ for all θ , then set $\theta_c = 0$.
 - Otherwise, compute θ_c using Eq. (H.16).
3. Compute the θ integrals in Eq. (H.17) using numerical quadrature (e.g. the trapezoidal rule). For a given θ , the value of r_w is determined numerically from

$$\sqrt{R^2 - r_w^2} = z_0 + \alpha(r_w^2 + 2hr_w \cos \theta)$$

if $\theta \leq \theta_c$, or

$$-\sqrt{R^2 - r_w^2} = z_0 + \alpha(r_w^2 + 2hr_w \cos \theta)$$

if $\theta > \theta_c$.

4. If $\mathcal{F} \neq 0$ then guess a new value of z_0 and repeat steps 2 and 3 until \mathcal{F} is sufficiently small. This process can be automated by using the secant method to solve $\mathcal{F}(z_0) = 0$ for z_0 .