

1994

Simulation of three-dimensional incompressible flows with free-surfaces including fluid-structure interaction and microgravity flows

Sethuraman Babu
Iowa State University

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

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Babu, Sethuraman, "Simulation of three-dimensional incompressible flows with free-surfaces including fluid-structure interaction and microgravity flows " (1994). *Retrospective Theses and Dissertations*. 10536.
<https://lib.dr.iastate.edu/rtd/10536>

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.

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.

UMI

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

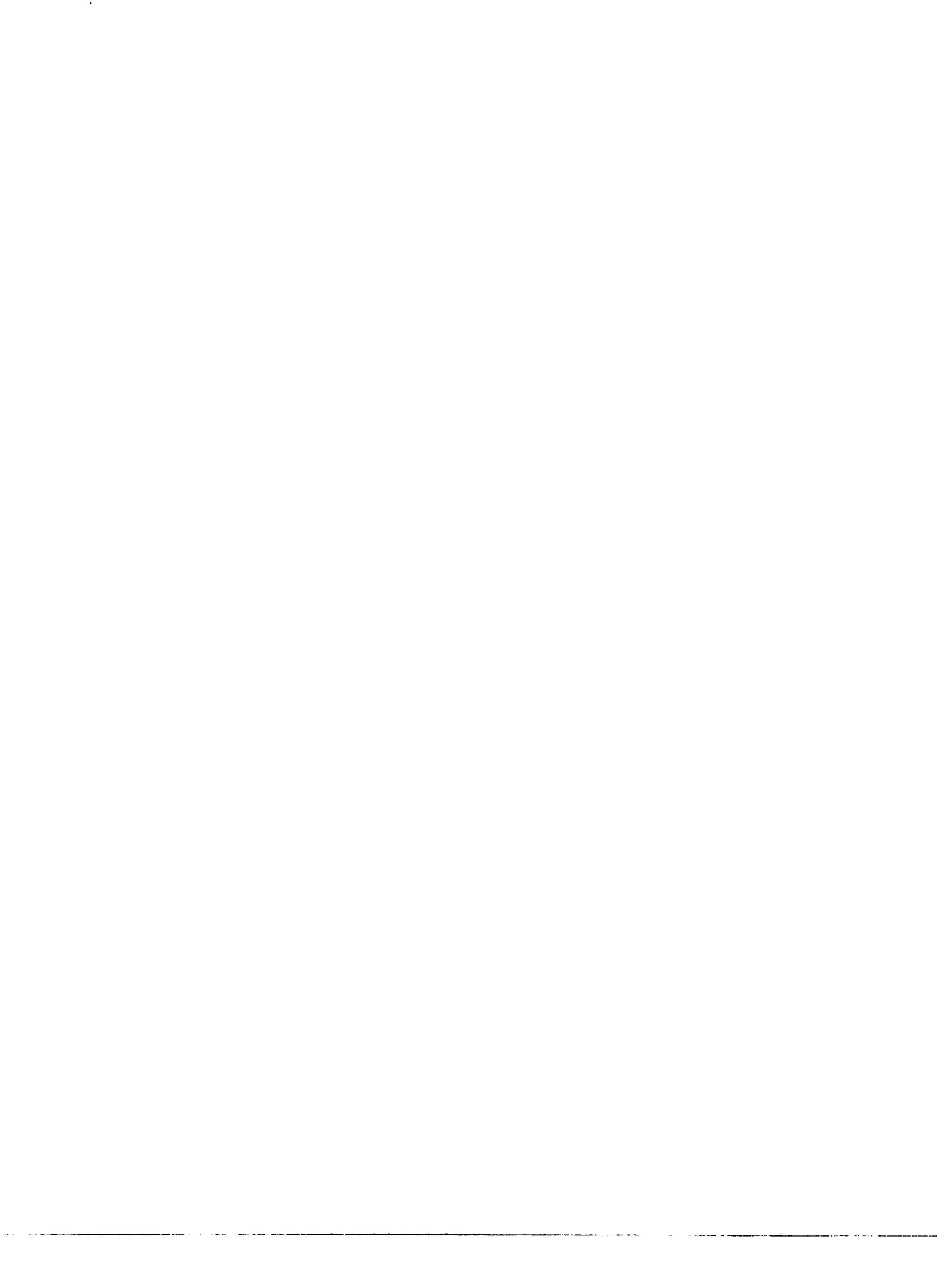
Order Number 9518356

Simulation of three-dimensional incompressible flows with free surfaces including fluid-structure interaction and microgravity flows

Babu, Sethuraman, Ph.D.

Iowa State University, 1994

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



**Simulation of three-dimensional incompressible flows with free-surfaces
including fluid-structure interaction and microgravity flows**

by

Sethuraman Babu

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
1994**

Copyright © Sethuraman Babu, 1994. All rights reserved.

TABLE OF CONTENTS

NOMENCLATURE	xi
ACKNOWLEDGMENTS	xiv
1. INTRODUCTION	1
1.1 Background	1
1.2 Literature review	2
1.3 Scope of this study	8
1.4 Thesis organization	9
2. MATHEMATICAL FORMULATION AND NUMERICAL PRO-	
 CEDURE	11
2.1 Sloshing inside moving containers	11
2.1.1 Governing equations	11
2.1.2 The elastic arm coordinates: x_2 coordinate system	16
2.1.3 Free-surface tracking coordinates	18
2.1.4 Nondimensionalization	20
2.1.5 Generalized transformation	21
2.1.6 Boundary conditions	22
2.2 Three-dimensional broken dam problem	25
2.2.1 Problem description	25

2.2.2	Governing equations	29
2.2.3	Boundary conditions	29
2.3	The artificial compressibility method	32
2.4	Discretization	33
2.4.1	Grid generation	36
2.5	Numerical procedure	38
2.5.1	Coupled strongly implicit procedure	41
2.6	Vectorization of the CSIP method	44
2.6.1	Vector processing	45
2.6.2	$[L][U]$ Decomposition	47
2.6.3	Planes of constant index sums	50
2.7	Smoothing	52
3.	LIQUID SLOSHING INSIDE SPHERICAL CONTAINERS . . .	54
3.1	Effects of vectorization	56
3.2	Code optimization	57
3.3	Contact line boundary condition	60
3.4	Free-surface motion: surface method 'B'	62
3.5	Results for the calculation of the orbital spin-up of a spherical tank	67
4.	VERIFICATION OF TRANSIENT ACCURACY OF THE SURFACE-FITTING APPROACH	87
4.1	The three-dimensional broken dam problem	87
4.2	Free-surface treatment: surface method 'C'	88
4.3	Discretization of the momentum equations	95
4.4	Results for the three-dimensional broken dam calculations	99

5. EFFECTS OF LOW GRAVITY	108
5.1 Discussion and Results	113
5.2 Calculations of zero gravity flows	118
6. FLUID-STRUCTURE INTERACTION	122
6.1 Introduction	122
6.2 Logic of interaction	124
6.3 Test case and results	127
6.4 Experiments	130
6.5 Numerical Simulation	132
6.5.1 Results and Discussions	133
6.6 Possible Improvements	138
7. CONCLUDING REMARKS	141
7.1 Conclusions	141
7.2 Suggestions for further research	142
BIBLIOGRAPHY	143
APPENDIX A. VECTOR COMPONENTS	153
APPENDIX B. LISTING OF CSIP FORTRAN ROUTINES	156
APPENDIX C. PERFORMANCE ANALYSIS	165
APPENDIX D. DIMENSIONAL ANALYSIS FOR THE SLOSH- ING MOTION OF LIQUIDS IN CYLINDRICAL CONTAIN- ERS ROTATING ABOUT THE AXIS	167

APPENDIX E. DERIVATION OF THE STEADY STATE FREE-SURFACE PROFILES FOR LIQUIDS INSIDE ROTATING CYLINDERS: CASES WITH THE SURFACE TOUCHING THE TOP WALL 169

LIST OF TABLES

Table 3.1:	Performance comparison of scalar and vector computer codes	60
Table 5.1:	Dimesionless parameters governing the axisymmetric sloshing motion inside cylinders	113
Table 5.2:	Comparison of the various parameters for the two cases . . .	114

LIST OF FIGURES

Figure 2.1:	Schematic of the satellite test rig	12
Figure 2.2:	Schematic of a rotating-nutating sphere	15
Figure 2.3:	Schematic of the three-dimensional broken dam problem experimental set up	27
Figure 2.4:	Liquid shape at different instances in the three-dimensional broken dam case	28
Figure 2.5:	Planes of constant index sums in three dimensional space . .	51
Figure 3.1:	Comparison of numerically computed in-board end free-surface transience and experimental data for different contact line boundary conditions	63
Figure 3.2:	The effect of number of free-surface updates on the global mass conservation	68
Figure 3.3:	Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank .	70
Figure 3.4:	Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank	73

Figure 3.5:	Transient motion of the inboard and outboard free-surface contact points, including analytical steady state values	76
Figure 3.6:	Comparison of the analytical and numerical steady state free-surface profile in the plane $x_{32} = 0$ for different number of free surface updates	77
Figure 3.7:	Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$	79
Figure 3.8:	Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$	82
Figure 3.9:	Transient motion of the inboard and outboard free-surface contact points, including analytical steady state values, $Re = 300$	85
Figure 3.10:	Comparison of the analytical and numerical steady state free-surface profile in the plane $x_{32} = 0$ for the $Re = 300$ case . .	86
Figure 4.1:	Examples of free-surface shapes that satisfy as well as that do not satisfy the single-valued requirement	88
Figure 4.2:	Schematic of the three-dimensional broken dam problem . . .	90
Figure 4.3:	Comparison of numerical and experimental transient data for the three-dimensional broken dam calculations	96
Figure 4.4:	Verification of grid independence of the broken dam free-surface calculations	97

Figure 4.5:	Comparison of experimental data on the transient surge front motion with numerical results from two different procedures .	100
Figure 4.6:	Position of flowing liquid at select instances	102
Figure 4.7:	Velocity vector plots in the $x_1 = 0$ plane at select instances .	104
Figure 4.8:	Velocity vector plots on $z_2 = \text{constant}$ planes at select instances	106
Figure 5.1:	Free-surface profiles at steady state	110
Figure 5.2:	Transient motion of the free-surface position at tank center .	111
Figure 5.3:	Comparison of transient motion of the free-surface position at tank center for the two cases	112
Figure 5.4:	Free-surface profiles at steady state, Impulsive start-up . . .	116
Figure 5.5:	Comparison of transient motion of the free-surface position at tank center for the initially capped and the impulsively started calculations	117
Figure 5.6:	Intermediate free-surface profiles during the transient motion from the initial condition to the steady state	120
Figure 5.7:	Free-surface profiles at steady state	121
Figure 6.1:	Logic of Interaction Program	125
Figure 6.2:	Explanation different modes of interaction	126
Figure 6.3:	Transient tank deflection data	128
Figure 6.4:	Transient free surface contact point motion	129
Figure 6.5:	Comparison of raw and smoothed transient angular velocity data	134

Figure 6.6: Comparison of numerical and experimental free-surface transient data 135

Figure 6.7: Comparison of numerical and experimental data on tank radial location 137

Figure 6.8: Comparison of numerical and experimental free-surface transient data, revised experiments 139

Figure 6.9: Comparison of numerical and experimental data on tank radial location, revised experiments 140

NOMENCLATURE

Roman Symbols

a	semi-width of column, broken dam problem
$[A]$	the coefficient matrix $[A]$ in the implicit procedure
$A_{i,j,k}^b$ etc.	submatrices of the coefficient matrix $[A]$
\vec{b}	right hand side vector
Bo	Bond Number
$CSIP$	coupled strongly implicit procedure
E_l	rotational quantity
Fr	Froude Number
f_{ij}	rotational quantities
F_1, F_2	free surface functions
\vec{g}	the gravity vector in
g_i	component of gravity in 'i' direction
h_i	length of elastic arm in 'i' direction
i_{max}	maximum value of index in z_1
j_{max}	maximum value of index in z_2
k_{max}	maximum value of index in z_3
k	iteration count

$[L]$	lower triangular matrix
n	time level count
\vec{n}	local unit normal vector
p	pressure
$[P]$	augmentation matrix in SIP method
\vec{q}	vector of primitive variables
r	radius of sphere
Re	Reynolds Number
s_{ij}	elements of matrix $[S]$
$[S]$	rotational transformation matrix
SIP	strongly implicit procedure
t	time
$[U]$	upper triangular matrix
\vec{V}	velocity vector
We	Weber Number
z_i	generalized transformed coordinate

Greek Symbols

α	partial cancellation parameter in SIP
α	grid clustering transformation constant
α_{ij}	elements of a transformation matrix
β	grid clustering transformation constant
$\beta_{i,li}$ etc.	rotational quantities

δ, Δ	finite differences
$\eta_{i,j}$	metric terms
$\eta_{i,jj}$	second order metric terms
ϕ_τ	angle of rotation for $[S]$ transformation
τ	generalized time
τ^*	pseudotime
ξ_i	rotation angles for the α transformation
ω	rotational velocity

Subscripts

i, j, k	grid point indices
ref	reference values

Superscripts

k	index associated with iteration counter
n	index associated with time level counter
0	associated with initial conditions

ACKNOWLEDGMENTS

I wish to extend my thanks to the Air Force Office of Scientific Research for their support of this research through Grant #AFOSR-89-0403. Thanks are also due to the personnel at NASA Lewis Research Center for providing access to the supercomputers at the Numerical Aerodynamic Simulation (NAS) facility, NASA Ames Research Center, Moffett Field, California.

I am greatly indebted to my major professor, Dr. Richard H. Pletcher, for his guidance and patience throughout this work. His suggestions and encouragement were very helpful in carrying out this research. I would also like to thank Drs. John Tannehill, Richard Hindman, Joseph Prusa, Patrick Kavanagh, Jon Van Gerpen and Ganesh Rajagopalan for serving on my examination committee. I should also thank many of my student friends and professors for their help during this research and my graduate study.

The encouragement shown by my parents Nagalakshmi and Sethuraman, my wife Geetha, my brother Anand and others in the family is sincerely appreciated. The personal support received from my mother during the final year at Iowa State certainly deserves special mention.

1. INTRODUCTION

1.1 Background

Many researchers have attempted to solve a variety of unsteady three-dimensional fluid flows. Until the last couple of decades, scientific attempts were confined to simpler flows: either two-dimensional or steady flows. Most flows arising in practical situations do not fall under either of these two classifications; hence the ability to compute three-dimensional unsteady flows is significant. With the onset of high speed digital computers, researchers have had the good fortune of being able to calculate more complex flows.

Recent successes at computing three-dimensional flows include calculations of flow over the space shuttle configuration [1] and the flow of blood inside body organs [4]. Besides being unsteady and three-dimensional, many flows possess additional complexities such as chemical reactions, turbulence, multiphase components, and the presence of discontinuities and interfaces. Even with the help of the most powerful computers available today, the computation of flows having more than a couple of these features is a very formidable task. Some of the limitations of present methods deal with factors such as stability, accuracy and computational efficiency. The need for improved and more efficient algorithms than those available currently to solve such problems should not be underestimated.

This research deals with flows that have a moving interface separating two fluids, generally a liquid and a gas. More specifically, the intention is to compute three-dimensional, unsteady, incompressible flows with one or more free-surfaces. The next section presents a review of relevant literature and the following section defines the scope of this study. The last section gives an outline of the various chapters that follow.

1.2 Literature review

Flow of air over an aircraft, of gases inside an internal combustion engine, and of water flowing over submerged bodies are examples of three-dimensional flows. The first two belong to the class of compressible flows and the last one is an incompressible flow. Due to the nature of such flows being very different from one another, so are the algorithms used to solve them. In general, the equations solved to obtain the solution of compressible flows are not used to solve incompressible flows. Due to the near infinite speed of sound, allowable time step sizes are severely restricted. This in turn results in the requirement of very large amounts of computer time for a calculation.

Incompressible flows can be solved in either primitive variables (like pressure and the three components of velocity) or in terms of derived variables like stream function and vorticity. The vorticity-stream function approach was one of the early methods used for solving incompressible flows. This method relies on splitting the mixed elliptic-parabolic system into separate elliptic and parabolic equations. This approach satisfies the continuity equation automatically [9] but necessitates solving a separate equation to obtain information on the pressure. The method is gener-

ally used only for two-dimensional cases and is not directly applicable for three-dimensional flows as there is no single stream function that satisfies the conservation of mass requirement. Other options such as the vorticity-velocity formulation [10] and the vorticity-potential formulation [12] generally require solving for more unknowns than the original Navier-Stokes equations. For three-dimensional applications, the primitive variable approach has seemed to be the most popular option.

In general, segregated calculation procedures for solving the incompressible Navier-Stokes equations do not generally satisfy the continuity equation automatically; these methods require a corrective scheme to adjust the velocity and pressure fields to satisfy the continuity equation. The class of schemes based on Patankar's SIMPLE method [8] fall under this category. More recently, a primitive variable formulation has been used to simultaneously solve for all the variables in such flows. The artificial compressibility method proposed by Chorin [20] was initially aimed at solving steady flows. This method was later put to use to solve unsteady flows [21] [22] [11]. Unsteady calculations of incompressible flows involve iterative solutions for the flow variables within a finite time step. As this step forms a significant portion of the overall calculations, acceleration of this process is generally desired and often implemented through various means [23].

The sloshing motion of liquids inside containers arises in many applications such as oil tankers, railroad tank cars, satellites and the space station. This phenomenon has been the subject of many previous investigations [19] [3] [5]. The liquid sloshing flows that are the focus of this research effort contain moving and deforming interfaces (free-surfaces), besides being generally unsteady and three-dimensional. The presence of a free-surface makes the problem more challenging as the position and shape of the

free-surface are not known beforehand and has to be evaluated as part of the solution. An excellent review of the various calculation procedures for viscous flows with free-surfaces has been done by Floryan and Rasmussen [48]. To briefly highlight some of these methods, they are divided into the following classes: Eulerian, Lagrangian or mixed. The present research uses an Eulerian procedure. The Eulerian methods can be either fixed grid or moving grid methods. This choice is an important decision that has to be made before the solution method is selected. There are specific advantages as well as disadvantages to both the approaches. These, and details of other methods, are discussed in the above mentioned review article. Nevertheless, two of the more important papers and their numerical methods are discussed below.

Harlow and Welch [56] used a method of marking fluid particles and tracking their motion to compute the overall flow. The physical space was divided into cells and those containing the particles were marked for the fluid presence. The governing equations were solved in the marked cells. The positions of the particles at the subsequent time step was updated through a Lagrangian method. This method was called the 'marker and cell' (MAC) method. The volume of fluid (VOF) method [57] devised by Hirt and Nichols identifies the amount of fluid in each cell. The function describing the amount of fluid in each cell varies with space as well as time, and its values ranges between '0' (meaning no fluid) and '1' (meaning filled with fluid). Cells with their function values which are some other fraction between 0 and 1 are identified to lie on the free surface. Both the marker and cell approach [30] as well as the volume of fluid method [31] [32] have subsequently been used to compute free-surface flows.

Another important decision that has to be made prior to the process of code

development relates to the method of solving the algebraic equations resulting from the discretization of the governing equations. Cramer's rule and Gauss-Siedel iteration were some of the early methods proposed for solving such systems of equations. Successive over relaxation, alternate direction implicit (ADI) schemes, followed in order. Stone [24] proposed a scheme for solving equations resulting from field problems. This approach has gained popularity in recent years [26] [27] and has undergone further improvements as well [25]. The present research uses the strongly implicit procedure of Stone, appropriately extended for coupled three-dimensional systems.

Computing resources available to researchers has been growing very rapidly over the last few decades. High speed workstations and large supercomputers have gained widespread use among computational fluid dynamics (CFD) researchers. Some of the computers that have come up recently are equipped with vector processors. In contrast to the processors that existed before, these possess the unique ability to operate simultaneously on multiple data. The ability to apply the same instruction to multiple data sets proves very useful in computationally intensive calculations and results in significant savings in computer time. Algorithms developed for scalar (unidata) processors are often not suitable for vector machines. The data dependencies arising from the implicit nature of these methods can prohibit vectorization.

Scientists are involved in a wide range of efforts including the search for better computer architectures and the evolution of better numerical algorithms, with the goal of improving the efficiency of physical process simulations. Researchers have found that vectorization of the solution algorithm could lead to significant speed-up of the numerical calculations [28]. Some studies using the SIP do mention the implementation of partial vectorization of the algorithm [29]. This thesis describes

in detail how the maximum vectorization of SIP is possible; these details have not been previously published to the best of the author's knowledge.

It is very important to verify the time accuracy of results from computations of unsteady flows. Verification of wall forces has been performed [33] for unsteady three-dimensional incompressible flows using an upwind procedure outlined in an earlier publication [21]. It should however be pointed out that this computation ignores viscosity effects. One of the difficulties associated with verifying transient free-surface flow results is the lack of adequate and reliable experimental data. It is for this reason that very few, if any, verifications of three-dimensional transient free-surface results have been published to date. Though the plots are not given, Harlow and Welch [56] suggest excellent agreement of their marker and cell (MAC) scheme results with experimental data. Similarly, although simulated flow pattern of sloshing liquid agrees well with observations [34], their transient accuracy is yet to be thoroughly verified. The transient results from the calculations in the current research are compared with experimental data for the three-dimensional broken dam problem.

The behavior of fluids, especially in situations with a free-surface, changes dramatically as the gravity levels are significantly reduced. In fact, multiple solutions of equilibrium free-surface shapes are possible when the shape of the container (surrounding the fluid) becomes exotic [35]. Careful treatment of the surface tension terms occurring in the boundary conditions is generally necessary in solving such problems. Modeling the surface tension as a localized volume force has been used in a study to compute the jet induced low gravity tank flow [36]. A series of low and microgravity calculations have been performed for different geometries using conven-

tional stress continuity conditions at the free-surface [68] - [70]. Similar boundary conditions are used in the low gravity calculations in this study as well.

The overall objective of the research project of which the present study was a part was to study the effect of liquid sloshing on the dynamics of spin stabilized satellites. Hence, it is important to estimate the effect of the liquid motion on the accompanying structure. As part of the Iowa State project, a finite element structural dynamics calculation procedure has been developed [37], [38]. This model is capable of treating individual components of the structure as either rigid or flexible bodies. This thesis describes the efforts to couple the CFD calculations and the structural dynamics calculations. This part of the study shall be referred to as fluid-structure interaction.

Numerous investigations have attempted as well as successfully performed some form of fluid-structure interactions. Calculations of forces of flowing liquids inside pipes, or over cylindrical arrays are some examples of early work of this nature. A recent article [39] has reviewed the research in this area and also discusses applications where they can become useful. Forces on a pair of spheres due to fluid flow around them, and the variations of these forces with the separation distance (between the spheres) has been the focus of another work [40]. Simulation of liquid filled containers subjected to an impulse (dropped under gravity) is another example that has applications in the transportation of hazardous or radioactive materials [41]. Fluid-structure interactive calculations, the results and their comparison with experimental data are also included in this thesis.

1.3 Scope of this study

Over the last several years, researchers at Iowa State University have been investigating the dynamics of spin-stabilized satellites carrying sloshing liquid stores [13] [14] [15] . As part of this project, the computational fluid dynamics (CFD) modeling of the sloshing motion of liquids has been attempted and their results reported earlier. The objective in all these cases has been to compute the motion of liquids in partially-filled containers subjected to arbitrary motion. The motion of liquids in such circumstances is governed by the three-dimensional incompressible Navier-Stokes equations and no further simplification can be made to the equations and still accurately represent the problem. Kassinos and Prusa [16] have attempted to solve the problem using a fractional step method and a derived Poisson equation for the pressure. Chen and Pletcher [17] [18] have approached the problem through the artificial compressibility method [20] and have solved the coupled system of equations using the strongly implicit procedure proposed by Stone [24] . This method updates all the primitive variables throughout the grid in one calculation sweep and thereby eliminates the need for solving a separate equation for the pressure. Both the above attempts place the free-surface at one boundary of the computational domain and calculate its motion by solving a kinematic condition. This method shall be referred to as the ‘surface-fitting’ approach in contrast to the ‘surface-capturing’ approach explained below. Efforts have also been aimed at calculating such flows by solving the governing equations in both fluids (generally a liquid and a gas) and ‘capturing’ the interface as a density discontinuity [42] .

The surface fitting approach seems to predict reasonably well for many flows. However this procedure encountered difficulties like poor mass conservation and in-

adequate stability at higher Reynolds numbers and larger free-surface deformation [17]. The surface capturing approach implemented by Kelecy also looks promising; but the initial version does not have the capability to account for surface tension effects [42]. Both of these issues are addressed as part of the present research.

The objective of the present study is to develop a time-accurate, computationally efficient divergence free algorithm for the prediction of the sloshing motion of liquids in containers. It is also intended to demonstrate the capability of the approach to calculate flows at microgravities or in situations where the surface tension forces are significant. Coupling the fluid flow computations with the structural dynamics calculations has always been one of the main aims of the overall project. This also forms part of the objectives of this research.

Several cases of sloshing motion of liquids in spherical and cylindrical containers, with gravity levels ranging from terrestrial to 10^{-5} g are computed and their results discussed.

1.4 Thesis organization

The next chapter details the formulation of the governing equations for this study. This deals with two parts: liquid sloshing in moving containers and the three-dimensional broken dam problem. Chapter three explains the vectorization of the strongly implicit procedure, an improved free-surface procedure (method 'B') and also discusses results from the calculation of the sloshing motion of liquid in orbiting spherical containers. Chapter four details the three-dimensional broken dam problem, the transient results for this case and how it compared with experiment.

Chapter five discusses some low gravity calculations and their results. Chapter

six discusses the procedure of implementing fluid-structure interaction. This chapter also includes results from the interactive calculations and compares them with experimental data. The last chapter lists the conclusions of this study and also points out areas for future research efforts.

2. MATHEMATICAL FORMULATION AND NUMERICAL PROCEDURE

2.1 Sloshing inside moving containers

The motion of liquids inside partially filled moving containers undergoing arbitrary motion is governed by the three-dimensional incompressible Navier-Stokes equations. As mentioned earlier, different methods do exist for solving these equations. The present research attempts to obtain the flow solution in terms of primitive variables using the artificial compressibility approach which allows the equations to be solved in a coupled manner. From among the different options described in the last chapter for handling the free-surface, the free-surface fitting approach was chosen for this present study. The algebraic equations resulting from the discretization are solved using the strongly implicit procedure. The rest of this chapter details the mathematical formulation of the problem and the numerical procedure used to solve the resulting system of equations.

2.1.1 Governing equations

As stated earlier, the motivation behind the CFD efforts is to compute the motion of liquids in satellite fuel stores. A typical calculation can be best explained with the help of a sketch of the test rig shown in Figure 2.1. The twin spherical tank

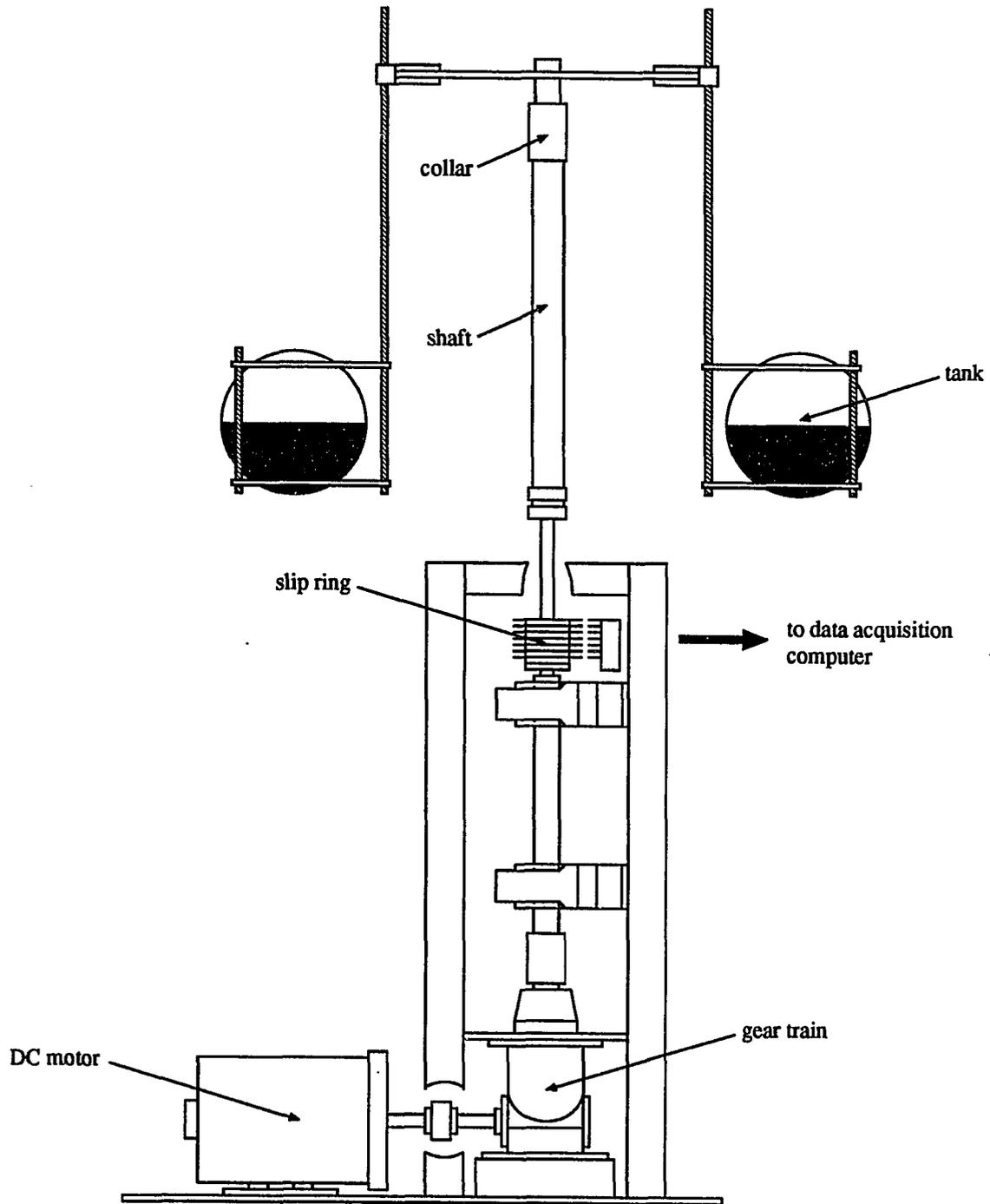


Figure 2.1: Schematic of the satellite test rig

system can be spun about the main vertical axis, and the horizontal bar supporting the spherical tanks can rotate about the point of attachment with the main vertical shaft.

Hence the tanks can undergo an arbitrary rotating-nutating motion. For the purpose of calculations, the tanks are generally assumed to be partially filled with a liquid; glycerine has been the liquid of choice for most calculations discussed in this thesis. The formulation of the governing equations and boundary conditions to compute the motion of the liquid in such a system has been explained in detail by Chen [17]. However, for the sake of completeness, the most important aspects of the formulation are repeated here. The motion of liquid is governed by the three-dimensional incompressible Navier-Stokes equations. These equations, for a Newtonian fluid, in primitive variable form written in index notation are:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2.1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - g_i \quad (2.2)$$

The first equation is the continuity equation and the second one stands for the three momentum equations where the index 'i' runs from 1 to 3. Occurrence of a repeated index in the same term represents a summation over three terms with the repeated index ranging from 1 to 3. The energy equation is not considered in solving the problems in this study as the variation of temperature is negligible. It is for this same reason that the properties like viscosity, μ , are also assumed to be constant throughout this thesis. In these equations, p stands for the pressure, u_i the three components of velocity, ρ the density, x_i the three coordinate directions and g for the

external gravity.

To facilitate easy application of boundary conditions, these equations are rewritten onto a non-inertial frame of reference whose origin coincides and moves with the geometric center of the spherical tank undergoing the rotating nutating motion. This is done through a series of rotational and translational transformations; in addition some steps involve definition of new relative velocities and non-dimensionalization. The following subsections detail the transformations involved and the evolution of the final form of the governing equations.

After each transformation, the relationship between the previous and present coordinate systems are written in the form of mathematical equations. The partial derivative expressions that form the chain rule relating the two coordinate systems are then listed. Using these chain rule expressions, the governing equations are then rewritten in the new coordinate system. This process continues until the desired form of the equations is obtained.

A schematic of the spherical tank undergoing a general rotating-nutating motion is given in Figure 2.2. The first transformation rewrites the equations on a coordinate system that is parallel to a coordinate system moving with the center of the spherical tank. This is achieved through a series of three successive rotations about the three axes; the rotational angles are chosen equal to the angle through which the sphere has rotated about that axis with reference to its initial position.

Mathematically, this transformation from the initial x_i frame to the x_{1i} coordinate system is given by:

$$x_i = \alpha_{ij}x_{1j} \tag{2.3}$$

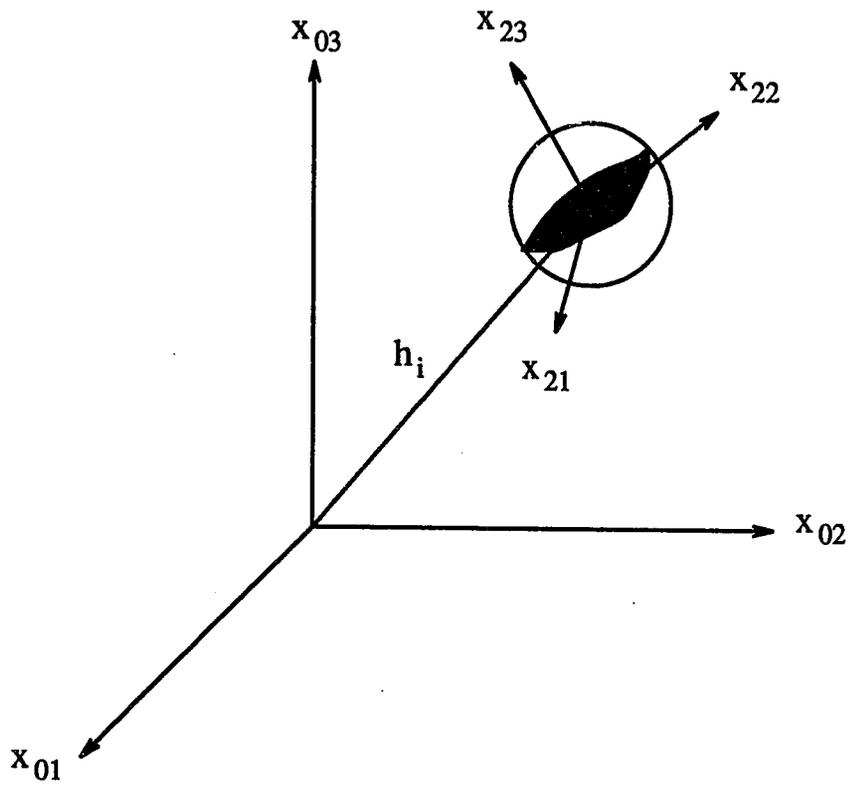


Figure 2.2: Schematic of a rotating-nutating sphere

where α_{ij} represents the elements of a 3×3 transformation matrix $[A]$ resulting from successive counterclockwise rotations through an angle of ψ_i about the three x_i axes respectively. This transformation matrix is given by:

$$[A] = \begin{bmatrix} C_2 C_3 & S_1 S_2 C_3 - C_1 S_3 & C_1 C_3 S_2 + S_1 S_3 \\ C_2 S_3 & S_1 S_2 S_3 + C_1 C_3 & C_1 S_2 S_3 - S_1 C_3 \\ -S_2 & S_1 C_2 & C_1 C_2 \end{bmatrix}$$

where $C_i = \cos \psi_i$ and $S_i = \sin \psi_i$.

2.1.2 The elastic arm coordinates: x_2 coordinate system

As the coordinate frame moving with the sphere and x_{1i} frame are now only parallel to each other, a translation through a distance of h_i is required to place them over one another.

This new coordinate frame shall be referred to as the x_2 frame and the relationship between x_1 and x_2 is:

$$x_{2i} = x_{1i} - h_i \quad (2.4)$$

Combining the relations between x_i and x_{1i} and that between x_{2i} and x_{1i} , the following relationship between x_i and x_{2i} can be obtained:

$$x_{2i} = \alpha_{ji} x_j - h_i \quad (2.5)$$

The inverse relation is given by:

$$x_i = \alpha_{ij} (x_{2j} + h_j) \quad (2.6)$$

Applying the chain rule to the derivative terms in Equations (2.1) and (2.2), the

following relationships between x_i and x_{2i} frames were obtained:

$$\begin{aligned}\frac{\partial}{\partial x_i} &= \alpha_{ij} \frac{\partial}{\partial x_{2j}} \\ \frac{\partial^2}{\partial x_i^2} &= \alpha_{ij} \alpha_{ik} \frac{\partial^2}{\partial x_{2j} \partial x_{2k}} = \delta_{jk} \frac{\partial^2}{\partial x_{2j} \partial x_{2k}} = \frac{\partial^2}{\partial x_{2j} \partial x_{2j}} \\ \frac{\partial}{\partial t} &= \frac{\partial}{\partial t_2} + [\dot{\alpha}_{ji} \alpha_{jk} (x_{2k} + h_k) - \dot{h}_i] \frac{\partial}{\partial x_{2i}}\end{aligned}$$

The governing equations in the x frame, i.e., Eqs. (2.1) and (2.2), can be written in terms of the new x_2 coordinates and new velocity components in the x_2 frame by using the above chain rule relationships. The new governing equations in the x_2 frame are:

$$\frac{\partial u_{2i}}{\partial x_{2i}} = 0 \quad (2.7)$$

$$\begin{aligned}\frac{\partial u_{2l}}{\partial t_2} - \dot{\alpha}_{il} \alpha_{ij} u_{2j} + [\dot{\alpha}_{ji} \alpha_{jk} (x_{2k} + h_k) - \dot{h}_i + u_{2i}] \frac{\partial u_{2l}}{\partial x_{2i}} \\ = -\frac{1}{\rho} \frac{\partial p}{\partial x_{2l}} + \nu \frac{\partial^2 u_{2l}}{\partial x_{2j} \partial x_{2j}} - g_{2l}\end{aligned} \quad (2.8)$$

where

$$u_{2i} = \alpha_{ji} u_j$$

$$g_{2l} = \alpha_{il} g_i$$

$$\dot{h}_i = \frac{dh_i}{dt}$$

$$\dot{\alpha}_{il} = \frac{d\alpha_{il}}{dt}$$

To further simplify the boundary condition implementation, a new relative velocity is defined in the x_2 frame. Once this is done, the velocity of fluid on the wall of the container shall always be zero no matter what type of motion the container undergoes. This new relative velocity is defined as:

$$u_{2l'} = u_{2l} + \dot{\alpha}_{ji}\alpha_{jk}(x_{2k} + h_k) - \dot{h}_i$$

The governing equations in x_2 frame can be rewritten in terms of the new relative velocity components by substituting the above expression in equations 2.7 and 2.8 .

The equations then turn out as:

$$\frac{\partial u_{2i}}{\partial x_{2i}} = 0 \quad (2.9)$$

$$\begin{aligned} \frac{\partial u_{2i}}{\partial t_2} + u_{2i} \frac{\partial u_{2l}}{\partial x_{2i}} - 2\beta_{cr,li}u_{2i} - \beta_{t,li}(x_{2i} + h_i) - \beta_{cp,li}(x_{2i} + h_i) \\ = -\frac{1}{\rho} \frac{\partial p}{\partial x_{2l}} + \nu \frac{\partial^2 u_{2l}}{\partial x_{2j} \partial x_{2j}} - g_{2l} + E_l \end{aligned} \quad (2.10)$$

where

$$\beta_{cr,li} = \dot{\alpha}_{jl}\alpha_{ji}$$

$$\beta_{t,li} = \ddot{\alpha}_{jl}\alpha_{ji}$$

$$\beta_{cp,li} = \dot{\alpha}_{kl}\dot{\alpha}_{ki} - \dot{\alpha}_{kl}\dot{\alpha}_{nj}\alpha_{kj}\alpha_{ni}$$

$$E_l = 2\dot{\alpha}_{il}\alpha_{ij}\dot{h}_j - \ddot{h}_l$$

$$\ddot{h}_l = \frac{d^2 h_l}{dt^2}$$

$$\ddot{\alpha}_{jl} = \frac{d^2 \alpha_{jl}}{dt^2}$$

In writing these equations, the l has been ignored for the sake of convenience.

2.1.3 Free-surface tracking coordinates

The classical free-surface boundary conditions used in this present study require the free-surface to be a single valued function in the independent variable domain.

The general free-surface shapes encountered in the sloshing problems may not satisfy this condition. Hence, another rotational transformation from the x_2 frame to the x_3 frame is required to ensure this. This transformation is achieved by effecting a counterclockwise rotation through an angle ϕ_r about the x_{22} axis. The relationship between the x_2 frame and the x_3 frame can be written as

$$x_{2i} = s_{ij}x_{3j} \quad (2.11)$$

where s_{ij} is an element of the transformation matrix $[S]$.

The transformation matrix $[S]$ is given by:

$$[S] = \begin{bmatrix} \cos \phi_r & 0 & -\sin \phi_r \\ 0 & 1 & 0 \\ \sin \phi_r & 0 & \cos \phi_r \end{bmatrix}$$

where ϕ_r is the angle through which the x_2 frame is rotated to get the x_3 frame of reference.

Applying the chain rule, the governing equations written in the x_3 frame are:

$$\frac{\partial u_{3i}}{\partial x_{3i}} = 0 \quad (2.12)$$

$$\begin{aligned} \frac{\partial u_{3n}}{\partial t_3} + (u_{3i} + f_{ij}x_{3j}) \frac{\partial u_{3n}}{\partial x_{3i}} - (f_{ni} + 2\lambda_{ni})u_{3i} - \tau_{1,ni}x_{3i} - \tau_{2,ni}h_i \\ = -\frac{1}{\rho} \frac{\partial p}{\partial x_{3n}} + \nu \frac{\partial^2 u_{3n}}{\partial x_{3j} \partial x_{3j}} - g_{3n} + E_n \end{aligned} \quad (2.13)$$

where

$$u_{3j} = s_{ij}u_{2i}$$

$$g_{3n} = s_{ln}g_{2l}$$

$$f_{ij} = \dot{s}_{ki} s_{kj}$$

$$\lambda_{ni} = \beta_{cr,ij} s_{ji} s_{ln}$$

$$\tau_{1,ni} = (\beta_{t,ij} + \beta_{cp,ij}) s_{ln} s_{ji}$$

$$\tau_{2,ni} = (\beta_{t,li} + \beta_{cp,li}) s_{ln}$$

$$E_n = s_{ln} E_l$$

2.1.4 Nondimensionalization

All the governing equations written so far contain dimensional variables. It is very convenient to nondimensionalize the various quantities so that their values fall within a specific range, like between 0 and 1. The nondimensional factors chosen are:

$$x_{3i}^* = \frac{x_{3i}}{L_{ref}} \quad u_{3i}^* = \frac{u_{3i}}{V_{ref}} \quad t^* = \frac{t}{t_{ref}} \quad p^* = \frac{p - p_0}{p_{ref}} \quad (2.14)$$

where the parameters with the *ref* subscript refer to reference quantities. These reference quantities differ from problem to problem and the parameters used for a specific problem shall be listed while discussing that case. The parameter p_0 stands for an appropriate pressure datum such as the atmospheric pressure or the partial pressure of the vapor above the free-surface.

Replacing the dimensional quantities in the governing equations according to the definitions given above leads to the following dimensionless form of the equations:

$$\frac{\partial u_{3i}}{\partial x_{3i}} = 0 \quad (2.15)$$

$$\frac{\partial u_{3n}}{\partial t_3} + (u_{3i} + f_{ij} x_{3j}) \frac{\partial u_{3n}}{\partial x_{3i}} - (f_{ni} + 2\lambda_{ni}) u_{3i} - \tau_{1,ni} x_{3i} - \tau_{2,ni} h_i$$

$$= -\frac{\partial p}{\partial x_{3n}} + \frac{1}{Re} \frac{\partial^2 u_{3n}}{\partial x_{3j} \partial x_{3j}} - g_{3n} + E_n \quad (2.16)$$

where Re is the Reynolds number and is defined as:

$$Re = \frac{V_{ref} L_{ref}}{\nu}$$

Again, in writing the above equations, the superscript $*$ has been omitted.

2.1.5 Generalized transformation

The nondimensional equations in the x_3 noninertial frame are still not best suited for handling irregular geometries. It is best to fit the irregular free-surface to a constant coordinate surface; this greatly simplifies boundary condition implementation. This is done by enforcing the following generalized transformation:

$$\tau = t_3 \quad (2.17)$$

$$z_i = z_i(x_{31}, x_{32}, x_{33}, t_3) \quad (2.18)$$

The partial derivatives are related by:

$$\begin{aligned} \frac{\partial}{\partial x_{3i}} &= \eta_{j,i} \frac{\partial}{\partial z_j} \\ \frac{\partial^2}{\partial x_{3i} \partial x_{3i}} &= \eta_{j,i} \eta_{k,i} \frac{\partial^2}{\partial z_j \partial z_k} + \eta_{k,ii} \frac{\partial}{\partial z_k} \\ \frac{\partial}{\partial t_3} &= \frac{\partial}{\partial \tau} + \dot{z}_i \frac{\partial}{\partial z_j} \end{aligned}$$

where

$$\eta_{i,j} = \frac{\partial z_i}{\partial x_{3j}},$$

$\eta_{i,jj} = \frac{\partial^2 z_i}{\partial x_{3j} \partial x_{3j}}$ are metric terms

and

$\dot{z}_i = \frac{\partial z_i}{\partial t_3}$ are the grid speed terms.

The final form of the governing equations in the generalized (z_i, τ) coordinates is given by:

$$\eta_{j,i} \frac{\partial u_{3i}}{\partial z_j} = 0 \quad (2.19)$$

$$\begin{aligned} \frac{\partial u_{3n}}{\partial \tau} + (\dot{z}_j + \eta_{j,i} u_{3i} + \eta_{j,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_j} - (f_{ni} + 2\lambda_{ni}) u_{3i} \\ - \eta_{j,n} \frac{\partial p}{\partial z_j} + \frac{1}{Re} (\eta_{j,i} \eta_{k,i} \frac{\partial^2 u_{3n}}{\partial z_j \partial z_k} + \eta_{k,ii} \frac{\partial u_{3n}}{\partial z_k}) \\ = -\tau_{1,ni} x_{3i} - \tau_{2,ni} h_i - g_{3n} + E_n, \end{aligned} \quad (2.20)$$

In these final equations, (τ, z_i) represents the generalized time and three space coordinates, u_{3i} the nondimensional velocity components; $\eta_{i,j}$ stand for the metric terms and \dot{z}_i the grid speed terms.

2.1.6 Boundary conditions

Two general classes of boundary conditions have been used in solving this problem: wall boundary conditions and free-surface boundary conditions. In case of the solid wall, no-slip conditions are used for the three velocities and a normal momentum equation (derived by a dot product of the three momentum equations and the unit normal at the wall) gives the pressure on the wall. For the case of free-surfaces, the continuity equation together with conditions for continuous stresses in three directions comprise the four conditions required for the primitive variables. In addition,

a kinematic boundary condition is used to calculate the motion of the free-surface after each time-step. Some special aspects of the boundary conditions, including the contact line (the portion of the free-surface that is in contact with the container wall) are discussed in further detail in a later chapter. To illustrate, some of the typical boundary conditions are described in the next few subsections.

2.1.6.1 Wall boundary conditions At the wall of the spherical container, zero relative motion between the fluid particles and the solid wall is specified through these three equations:

$$u_{3i} = 0 \quad i = 1, 2, 3 \quad (2.21)$$

The equation for pressure, representing the conservation of momentum in the direction normal to the wall, is formed by performing the inner product of the *local* unit vector normal to the wall and the three momentum equations. The three components of the *local* unit normal vector at the wall are given by:

$$n_i = -\frac{\eta_{2,i}}{(\eta_{2,j}^2)^{1/2}} \Big|_{\text{at wall}}$$

where \hat{e}_{31} , \hat{e}_{32} and \hat{e}_{33} are the unit base vectors in the x_{31} , x_{32} and x_{33} directions respectively.

An inner product of this vector and the momentum equations is represented by:

$$M_n = \vec{n} \cdot \vec{M}$$

where M_n represents the normal momentum equation, \vec{n} is the *local* unit normal vector at the wall and \vec{M} represents the three momentum equations in a vector form.

The normal momentum equation at the wall, derived through such an inner product is given by:

$$\begin{aligned} \eta_{j,n}\eta_{2,n}\frac{\partial p}{\partial z_j} = & \{\tau_{1,ni}x_{3i} + \tau_{2,ni}h_i - g_{3n} + E_n - (\dot{z}_2 + \eta_{2,i}f_{ik}x_{3k})\frac{\partial u_{3n}}{\partial z_2} \\ & + \frac{1}{Re}[\eta_{2,i}^2\frac{\partial^2 u_{3n}}{\partial z_2^2} + \eta_{2,ii}\frac{\partial u_{3n}}{\partial z_2} + 2\eta_{1,i}\eta_{2,i}\frac{\partial^2 u_{3n}}{\partial z_1\partial z_2} + 2\eta_{2,i}\eta_{3,i}\frac{\partial^2 u_{3n}}{\partial z_2\partial z_3}]\}\eta_{2,n} \end{aligned} \quad (2.22)$$

In the above equation, it is assumed that the wall of the container coincides with the $z_2 = \text{constant}$ surface in Eq. (2.22). This equation is valid only for the sloshing in spherical containers. For the three dimensional broken dam problem and the sloshing in cylindrical containers, the wall of the container does not always coincide with the $z_2 = \text{constant}$ surface. These equations take on a slightly different form and they are discussed later at appropriate situations.

2.1.6.2 Free-surface boundary conditions The following equations comprise the boundary conditions used:

1. *Continuity equation*

$$\eta_{j,i}\frac{\partial u_{3i}}{\partial z_j} = 0 \quad (2.23)$$

2. *Zero tangential shear stress(two equations)*

$$\frac{\partial U_n}{\partial \tau_1} + \frac{\partial U_{\tau_1}}{\partial n} + U_{\tau_1}\kappa_1 = 0 \quad (2.24)$$

$$\frac{\partial U_n}{\partial \tau_2} + \frac{\partial U_{\tau_2}}{\partial n} + U_{\tau_2}\kappa_2 = 0 \quad (2.25)$$

3. Continuous normal shear stress

$$p - \frac{2}{Re} \frac{\partial U_n}{\partial n} - \frac{1}{We} \kappa = 0 \quad (2.26)$$

4. Free surface kinematic condition

$$\begin{aligned} \frac{\partial F}{\partial \tau} &= \{u_{33} + f_{3k}x_{3k}\} \\ &\quad - \{\dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2}\} \frac{\partial F}{\partial z_1} \\ &\quad - \{\dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2}\} \frac{\partial F}{\partial z_2} \end{aligned} \quad (2.27)$$

where F , the free-surface height is a function of time, and spatial coordinates x_{31} and x_{32} . In the above free-surface kinematic condition, the surface is assumed to coincide with the $z_3 = \text{constant}$ surface. For the three-dimensional broken dam calculation, the free-surface boundary conditions are listed in the next section.

2.2 Three-dimensional broken dam problem

2.2.1 Problem description

During the atomic bomb tests conducted in the first half of this century, it was observed that a large explosion in shallow water throws a large quantity of water vertically upward and forms a column. This column, consisting of a mixture of fine droplets of water and air, collapses soon after the explosion causing a rapid spreading from the base. In the early 1950's interest in the phenomenon of collapsing liquid columns grew stemming mainly from a desire to better understand the physics of 'base

surges' [53] [54]. As part of a project, scientists at Oxford University, UK performed analytical as well as experimental investigation of the collapse of fluid columns [52]. The broken dam experimental results from their study have been used to verify the transient accuracy of many free-surface solvers including the present solution scheme. This section briefly describes the experimental set up as a means of introducing the problem.

In a typical experiment, a fluid column, initially at rest on a rigid horizontal plane, is allowed to drop; the fluid spreads out on the supporting horizontal rigid plane and the height of the column falls. Air was generally the lighter fluid surrounding the heavier liquid although the experiments were intended to support theoretical studies that assumed *in vacuo* conditions. Columns of various cross sections, cylindrical as well as rectangular, were studied. The liquid column was initially held in place by a thin diaphragm on one side and solid transparent perspex walls on all the others sides. Figure 2.3 shows one such set up for a rectangular column, which happened to be the one used to verify the present code. The dimensions of the tank were $2\frac{1}{4} \times 2\frac{1}{4} \times 5$ inches. The thin paper diaphragm could be destroyed and the liquid allowed to flow by manually closing a switch that in turn send a heavy electric current through the diaphragm. A perspex channel of slightly greater width than the reservoir column was used to guide the outflowing liquid.

The motion of the fluid as the surge front progressed along the channel, as well as the residual column that dropped continuously, was recorded by a relatively high speed (300 frames per second) camera. Timing marks from a standard oscillator were also recorded on the film simultaneously. The progression of the fluid front and also the residual column could then be estimated by comparing with previously

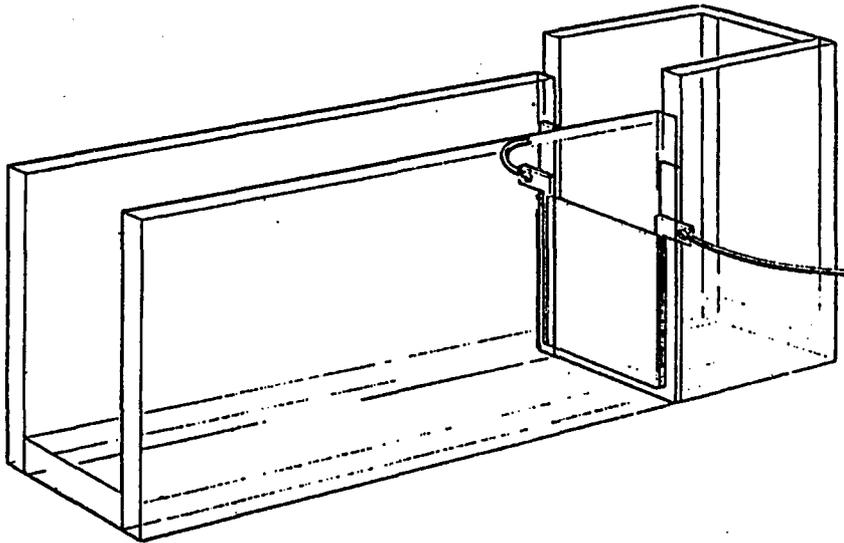


Figure 2.3: Schematic of the three-dimensional broken dam problem experimental set up

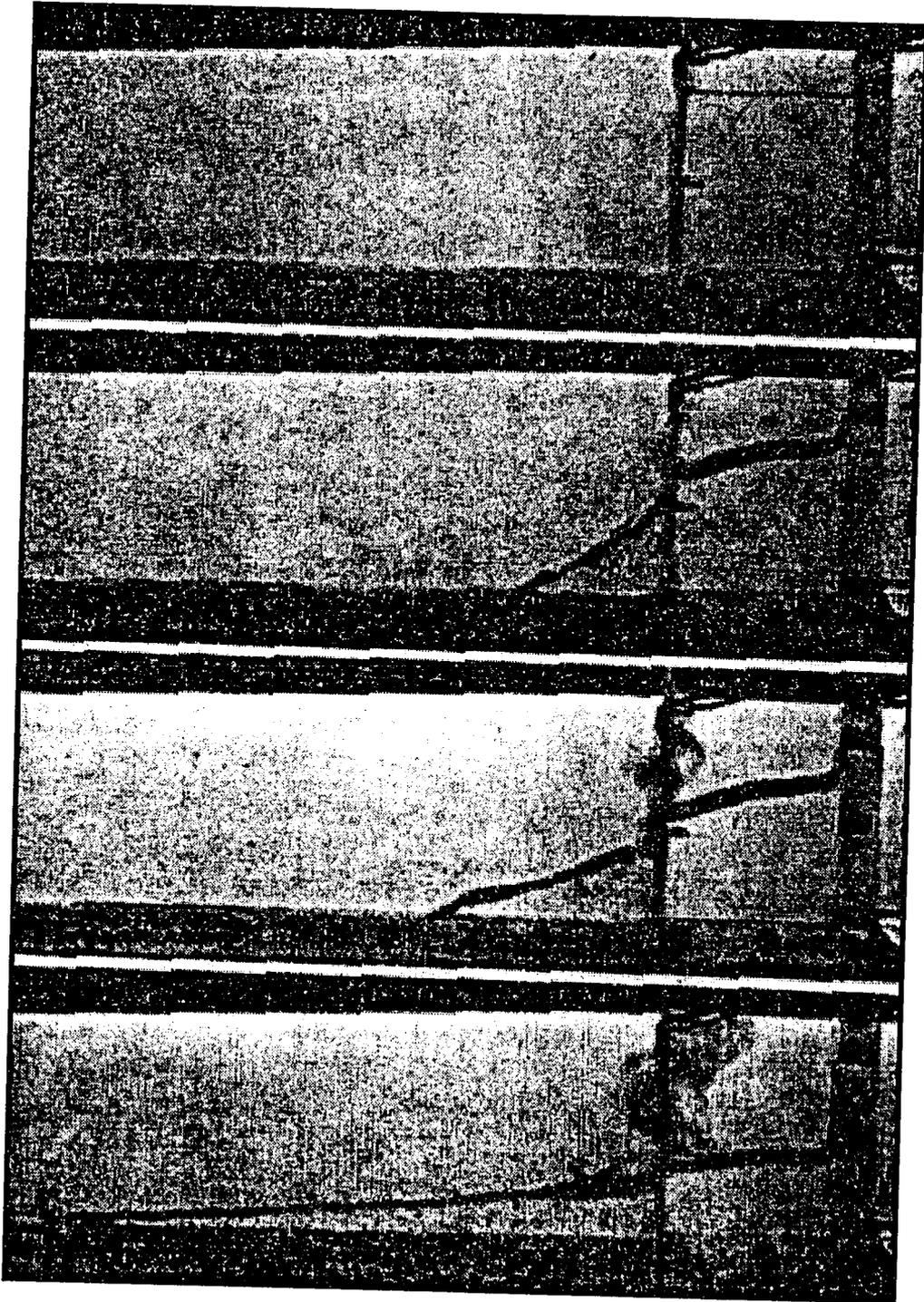


Figure 2.4: Liquid shape at different instances in the three-dimensional broken dam case

scored marker lines at the tank bottom and sides. Thus, data of the form 'surge front position vs time' and 'height of residual column vs time' were available for comparison. Figure 2.4 shows the liquid column in its initial state and a typical later instance. The following subsections describe the equations and the boundary conditions used for this particular problem.

2.2.2 Governing equations

In this problem, the container was stationary. Hence, its rotational velocities and accelerations are zero; the elastic arm length is also zero. This means that some of the terms in the governing equations 2.24 vanish. Instead of using the different set of equations, the same equations were solved after explicitly setting the appropriate terms to zero in the actual computations. This was done in order to be able to use the same computer code rather than make modifications to it. The fact that some of the inputs namely, ψ_i , and ϕ_r were zero in turn result in the corresponding transformation matrices $[A]$ and $[S]$ being equal to identity matrices.

2.2.3 Boundary conditions

The main difference in the equations used for the broken dam problem compared to the liquid sloshing cases is in the boundary conditions. The implementation of both the wall as well as the free-surface boundary conditions are different from the treatment adopted for the sloshing calculations. The distinction between the boundary conditions employed for the two types of computations are reviewed next.

2.2.3.1 Wall boundary conditions The conditions for the velocities at the wall are the no slip conditions similar to the ones used for the liquid sloshing calculations. These conditions specify that there is no relative motion between the fluid and the container wall at the wall. However, the conditions used for the pressure are different from the ones used for the sloshing problem. The container wall does not always coincide with $z_2 = \text{constant}$ surface in the case of the broken dam. The wall is composed of parts that are defined by $z_i = \text{constant}$ where the index 'i' runs from 1 to 3. The equation for the pressure on the wall is different for each constant z_i surface. The unit normals for each of these walls and the corresponding normal momentum equation are given below:

1. $z_1 = \text{constant}$

(a) Unit normal:

$$\vec{n} = n_i \hat{e}_{3i}$$

where

$$n_i = - \left[\frac{\eta_{1,i}}{\eta_{1,j}^2 \frac{1}{2}} \right]_{\text{wall}}$$

(b) Normal momentum equation:

$$\begin{aligned} \eta_{j,n} \eta_{1,n} \frac{\partial p}{\partial z_j} = & \{ \tau_{1,ni} x_{3i} + \tau_{2,ni} h_i - g_{3n} + E_n - (\dot{z}_1 + \eta_{1,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_1} \\ & + \frac{1}{Re} [\eta_{1,i}^2 \frac{\partial^2 u_{3n}}{\partial z_1^2} + \eta_{1,ii} \frac{\partial u_{3n}}{\partial z_1} + 2\eta_{1,i} \eta_{2,i} \frac{\partial^2 u_{3n}}{\partial z_1 \partial z_2} + 2\eta_{1,i} \eta_{3,i} \frac{\partial^2 u_{3n}}{\partial z_1 \partial z_3}] \} \eta_{1,n} \end{aligned} \quad (2.28)$$

2. $z_2 = \text{constant}$

(a) Unit normal:

$$\vec{n} = n_i \hat{e}_{3i}$$

where

$$n_i = - \left[\frac{\eta_{2,i}}{\eta_{2,j}^2 \frac{1}{2}} \right]_{wall}$$

(b) Normal momentum equation:

$$\begin{aligned} \eta_{j,n} \eta_{2,n} \frac{\partial p}{\partial z_j} &= \{ \tau_{1,n} x_{3i} + \tau_{2,n} h_i - g_{3n} + E_n - (\dot{z}_2 + \eta_{2,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_2} \\ &+ \frac{1}{Re} [\eta_{2,i}^2 \frac{\partial^2 u_{3n}}{\partial z_2^2} + \eta_{2,ii} \frac{\partial u_{3n}}{\partial z_2} + 2\eta_{1,i} \eta_{2,i} \frac{\partial^2 u_{3n}}{\partial z_1 \partial z_2} + 2\eta_{2,i} \eta_{3,i} \frac{\partial^2 u_{3n}}{\partial z_2 \partial z_3}] \} \eta_{2,n} \end{aligned} \quad (2.29)$$

3. $z_3 = \text{constant}$

(a) Unit normal:

$$\vec{n} = n_i \hat{e}_{3i}$$

where

$$n_i = - \left[\frac{\eta_{3,i}}{\eta_{3,j}^2 \frac{1}{2}} \right]_{wall}$$

(b) Normal momentum equation:

$$\begin{aligned} \eta_{j,n} \eta_{3,n} \frac{\partial p}{\partial z_j} &= \{ \tau_{1,n} x_{3i} + \tau_{2,n} h_i - g_{3n} + E_n - (\dot{z}_3 + \eta_{3,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_3} \\ &+ \frac{1}{Re} [\eta_{3,i}^2 \frac{\partial^2 u_{3n}}{\partial z_3^2} + \eta_{3,ii} \frac{\partial u_{3n}}{\partial z_3} + 2\eta_{1,i} \eta_{3,i} \frac{\partial^2 u_{3n}}{\partial z_1 \partial z_3} + 2\eta_{2,i} \eta_{3,i} \frac{\partial^2 u_{3n}}{\partial z_2 \partial z_3}] \} \eta_{3,n} \end{aligned} \quad (2.30)$$

2.2.3.2 Free-surface boundary conditions In the modeling of liquid sloshing problems, there was only one free-surface; the analysis for the broken dam problem allows the free-surface to be decomposed into two or more patches. Solution for each free-surface patch requires solving a separate kinematic condition. The free-surface kinematic equations used for this problem are described in a chapter four along with the discussions on their solution procedure.

The other four boundary conditions used at the free-surface, namely the three continuous stress conditions and the continuity equation are the same as for the liquid sloshing case. The unit normal and tangent vectors at the free-surface take on different forms for the different free-surface shape definitions. The components of these vectors are derived in Appendix A.

2.3 The artificial compressibility method

The set of governing equations given by Equation 2.20 are not best suited for discretization and further numerical procedures in the present form. This is due to the invariant density which consequently weakens the pressure-velocity coupling in the system of equations. In the absence of proper treatment, this leads to a velocity field that satisfies the momentum equations, but is not divergence free. In other words, it is possible to get solutions that satisfy the momentum equations but not the continuity equation.

In the previous chapter, methods used to overcome this difficulty were outlined. The option chosen for this study uses the artificial compressibility method. In this approach, a pseudo-time derivative of pressure of the form $\frac{\partial p}{\partial \tau^*}$ is added to the continuity equation. In a transient calculation, subiterations in the pseudo-time frame

drive the divergence to zero in every physical time step, $\delta\tau$. The equations with this ‘artificial compressibility’ term added take the form:

$$\frac{\partial p}{\partial \tau^*} + \eta_{j,i} \frac{\partial u_{3i}}{\partial z_j} = 0 \quad (2.31)$$

and

$$\begin{aligned} \frac{\partial u_{3n}}{\partial \tau} + (\dot{z}_j + \eta_{j,i} u_{3i} + \eta_{j,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_j} - (f_{ni} + 2\lambda_{ni}) u_{3i} \\ + \eta_{j,n} \frac{\partial p}{\partial z_j} - \frac{1}{Re} (\eta_{j,i} \eta_{k,i} \frac{\partial^2 u_{3n}}{\partial z_j \partial z_k} + \eta_{k,ii} \frac{\partial u_{3n}}{\partial z_k}) \\ = \tau_{1,ni} x_{3i} + \tau_{2,ni} h_i - g_{3n} + E_n \end{aligned} \quad (2.32)$$

2.4 Discretization

The final form of the governing equations define the conservation laws in the entire liquid region. The solution to this system consists of dependent variables that are defined everywhere in the problem domain. In most numerical methods, this solution is replaced by solutions (dependent variables) defined only at discrete points. The partial derivatives forming the governing equations are then expressed in terms of finite difference expressions involving various parameters at these discrete locations. Thus the partial difference equations are cast into a system of algebraic equations.

The discretization used for the various terms in the governing equations is given below. These expressions are for a typical interior node (i, j, k) and a missing index denotes that the parameter is unremoved from the specific $i, j, \text{ or } k$ location.

1. Unsteady term

Terms of the type $\frac{\partial q}{\partial \tau}$ were replaced as follows:

$$\left[\frac{\partial q}{\partial \tau} \right]^{n+1} = \frac{q^{n+1} - q^n}{\Delta \tau}$$

Here, 'q' stands for any dependent variable, subscripts denote spatial locations and superscripts stand for the time level. Such a backward difference is generally considered first-order accurate.

2. Pseudo-time term

The artificial compressibility term, $\frac{\partial p}{\partial \tau^*}$, added to the continuity equation is differenced in the following way:

$$\frac{\partial p}{\partial \tau^*} = \frac{p^{n+1,k+1} - p^{n+1,k}}{\Delta \tau^*}$$

where the superscript 'n' stands for the time level, and 'k' for the pseudo-time iteration counter. These pseudo-time iterations are referred to as 'subiterations' in the physical time step. This aspect is explained in the next section.

3. Convective term

These terms, also referred to as the inviscid flux terms, are of the type $\frac{\partial q}{\partial z_1}$.

These terms were approximated by central differencing such as:

$$\left[\frac{\partial q}{\partial z_1} \right]^{n+1} = \frac{q_{i+1}^{n+1} - q_{i-1}^{n+1}}{2}$$

Similar central differences were used for fluxes in other directions as well. For example,

$$\left[\frac{\partial q}{\partial z_2} \right]^{n+1} = \frac{q_{j+1}^{n+1} - q_{j-1}^{n+1}}{2}$$

It should be noted that the points in discrete space were defined to be at unit distance intervals apart in all three generalized space directions. This standard practice makes programming very easy. Consequently, all quantities of the form Δz_i are equal to unity and hence are not shown whenever they occur as multiplying constants in the finite difference expressions.

4. Viscous term

Second order central differences were used to approximate such terms as:

$$\left[\frac{\partial^2 q}{\partial z_1^2} \right]^{n+1} = q_{i+1}^{n+1} - 2q_i^{n+1} + q_{i-1}^{n+1}$$

$$\left[\frac{\partial^2 q}{\partial z_2^2} \right]^{n+1} = q_{j+1}^{n+1} - 2q_j^{n+1} + q_{j-1}^{n+1}$$

5. Source term

Source terms were evaluated at the specified grid location as far as possible. Whenever an exception was made to this rule, it has been explained in the specific problem description.

6. Second order cross derivative term

These terms are of the type $\frac{\partial}{\partial z_1} \left(\frac{\partial q}{\partial z_2} \right) = \frac{\partial^2 q}{\partial z_1 \partial z_2}$, and were approximated in the following fashion:

$$\frac{\partial^2 q}{\partial z_1 \partial z_2} = \frac{q_{i+1,j+1}^{n+1} - q_{i+1,j-1}^{n+1} - q_{i-1,j+1}^{n+1} + q_{i-1,j-1}^{n+1}}{4}$$

and

$$\frac{\partial^2 q}{\partial z_1 \partial z_3} = \frac{q_{i+1,k+1}^{n+1} - q_{i+1,k-1}^{n+1} - q_{i-1,k+1}^{n+1} + q_{i-1,k-1}^{n+1}}{4}$$

2.4.1 Grid generation

A grid that defines points as intersections of lines belonging to two different families is classified as structured grid. In such a grid, a one-to-one mapping of points within an arbitrary region to an equal number of points forming a regular shape is possible. A random distribution of points in the physical space, with the points identified by their connectivity with their neighbors constitutes an unstructured grid. Unstructured grids possess the advantage of easily adapting to complex geometries, but become numerically complicated to process especially in three-dimensional calculations. A structured grid has been used in this study.

The computational strategy in the present surface fitting method forces the free-surface to be part of the boundary of the computational domain. The general irregular shape of the liquid is then transformed into a cubic shape in the computational space. The grid points in the computational space (and hence its corresponding point in the physical domain) can be identified uniquely by a set of three indices namely (i, j, k) . Points on the boundary of the liquid region (those on the free-surface as well as on the container wall) are placed on the computational space. Similarly, points in the interior of the liquid are mapped onto points inside the computational domain.

2.4.1.1 Grid generation for liquid sloshing in containers The grid generation method for the spherical sloshing computations has been explained in detail by Chen [17]. Points in the circumferential direction are generated with uniform spacing, and points in the radial and the height directions are generated algebraically using clustering techniques. The same approach has been used in this study for the calculation of sloshing in spherical containers. For the case of sloshing in cylindrical containers, the method differs only very slightly. These differences have been highlighted in the chapter on low gravity flows.

2.4.1.2 Grid generation for the broken dam problem The grid generation for the broken dam calculations have been done through algebraic methods. Predetermined constants were used to fix the relative position of points in each of the three directions. In the '*i*' direction (normal to the principal flow) the planes of constant z_1 are chosen through the following equation:

$$x = x_{max} \frac{(\beta + 2\alpha)[(\beta + 1)/(\beta - 1)]\exp[(x^* - \alpha)/(1 - \alpha)] - \beta + 2\alpha}{(2\alpha + 1)\{1 + [(\beta + 1)/(\beta - 1)]\exp[(x^* - \alpha)/(1 - \alpha)]\}} \quad (2.33)$$

The parameter α was set to 0.5 giving rise to equal clustering near both the side walls of the channel. Prior to the beginning of each grid generation sequence, the free-surface functions, $F_1(x_{31}, x_{32}, \tau)$ and $F_2(x_{31}, x_{33}, \tau)$ are known. These functions in turn give the value of $x_{33}(i, j, kmax)$ and $x_{32}(i, jmax, k)$, respectively. The index values $jmax$ and $kmax$ stand for the maximum values of j and k respectively. Using these data, the x_{32} and x_{33} values at other grid locations are found using the following expressions:

$$y = y_{max} \frac{(\beta + 2\alpha)[(\beta + 1)/(\beta - 1)]\exp[(y^* - \alpha)/(1 - \alpha)] - \beta + 2\alpha}{(2\alpha + 1)\{1 + [(\beta + 1)/(\beta - 1)]\exp[(y^* - \alpha)/(1 - \alpha)]\}} \quad (2.34)$$

$$z = z_{max} \frac{(\beta + 2\alpha)[(\beta + 1)/(\beta - 1)]\exp[(z^* - \alpha)/(1 - \alpha)] - \beta + 2\alpha}{(2\alpha + 1)\{1 + [(\beta + 1)/(\beta - 1)]\exp[(z^* - \alpha)/(1 - \alpha)]\}} \quad (2.35)$$

where the 'max' values correspond to the free-surface heights and the '*' values correspond to the transformed coordinate values. These equations are in fact the inverses of the Roberts type transformations: [50]

$$y^* = \alpha + (1 - \alpha) \frac{\ln(\{\beta + [y(2\alpha + 1)/y_{max}] - 2\alpha\}/\{\beta - [y(2\alpha + 1)/y_{max}] + 2\alpha\})}{\ln[(\beta + 1)/(\beta - 1)]} \quad (2.36)$$

In these transformations, a value of $\alpha = 0$ clusters the mesh near $y = y_{max}$ alone, and a value of $\alpha = 0.5$ refines the mesh equally at both ends, $y = 0$ and $y = y_{max}$. In the broken dam problem, and in most other free-surface flows considered throughout this thesis, grid refinement was found to be more critical near the free-surface. Hence, generally α was set to zero.

Thus, the equations given above define the grid for the entire three-dimensional space. This procedure essentially generates interior lines (planes) that conform to the shape of the free-surface(s).

2.5 Numerical procedure

A general transient calculation consists of marching the solution in time. Assuming all the required information is known at a specific instant τ , the following

procedure is used to advance the solution through a fixed physical time step $\delta\tau$. At the end of this cycle, all the relevant information is known at this later time, $\tau + \delta\tau$. This cycle consists of various steps: estimating free surface position at time $\tau + \delta\tau$, generating a grid and calculating metric and grid speed quantities, estimating the matrix $[A]$ and vector \vec{b} , and subiterating for the convergence of primitive variables.

Since its initial implementation, the free-surface procedure [17] has been revised considerably. A major aspect of this thesis involves description of this gradual evolution. In all, three different versions of the free-surface calculation procedure exist. The first of these, ‘surface method A’, represents the procedure developed by Chen. The next one, called ‘surface method B’ was used in the liquid sloshing in spherical and cylindrical containers. This method is described in detail in the next chapter. The final and the most advanced free-surface calculation procedure is called ‘surface method C’. This method has been found to be superior compared to the other two methods. The broken dam calculations used this procedure and the explanation of this method ‘C’ can be found in chapter four.

The various steps of method ‘A’, in a typical transient calculation are listed below. The purpose of describing this procedure here is twofold. It serves as a means of highlighting the improvements of the method to be described later. Besides, it helps identify the role and position of the coupled strongly implicit procedure (CSIP) in a transient free-surface calculation; a description of the CSIP algorithm is given in the next section. The steps involved in a typical transient calculation using the surface method ‘A’ are:

Known: All the primitive variables ($u_{31}, u_{32}, u_{33}, p$) at time τ (time level n).

1. Find free surface position at time level $n + 1$ by solving the kinematic condition.
2. Generate a grid; estimate metrics and grid speed quantities at this time level, $n + 1$
3. Calculate the elements of matrix $[A]$ and the RHS vector \vec{b}
4. By passing through CSIP once (one subiteration), generate the incremental change in the primitive variable vector $\delta\vec{q}$; update the vector \vec{q}
5. Check for convergence within this time level; if converged, go to the next step; if not converged, repeat steps 3 and 4 until convergence is achieved within this time step.
6. Check if the calculations have progress to the desired time or to steady state; if not, increment time level counter and go to step 1. If it has, the calculations can be stopped.

The criterion for convergence within the present time level requires that the change from one subiteration to the next, $\frac{p^{n+1,k+1} - p^{n+1,k}}{\Delta\tau^*}$ which is an approximation for the divergence, be smaller than a predetermined value. That is,

$$\frac{p^{n+1,k+1} - p^{n+1,k}}{\Delta\tau^*} \leq \epsilon_1$$

while a typical value of ϵ_1 used was 8.0×10^{-4} . The criterion for achieving steady state required that the maximum change in the norm of the primitive variable vector from one time step to the next be smaller than a small value. Mathematically, this can be expressed as:

$$\left| \sqrt{\left\{ \frac{(p^2 + q_{31}^2 + q_{32}^2 + q_{33}^2)}{4} \right\}_{n+1}} - \sqrt{\left\{ \frac{(p^2 + q_{31}^2 + q_{32}^2 + q_{33}^2)}{4} \right\}_n} \right|_{\max} \leq \epsilon_2$$

Values of ϵ_2 used were of the order of 1.0×10^{-9} . Detailed description of the coupled strongly implicit procedure that is used to update the primitive variable vector through subiterations at a specific time level is given in the next few pages.

2.5.1 Coupled strongly implicit procedure

Once the governing equations have been discretized and expressed at various points in a grid, the next step involves solving the resulting system of algebraic equations. The strongly implicit procedure, first introduced by Stone has been used in all the calculations to solve the algebraic system of equations. A typical equation looks like:

$$\begin{aligned} A_{i,j,k}^b \vec{q}_{i,j,k-1} + A_{i,j,k}^s \vec{q}_{i,j-1,k} + A_{i,j,k}^w \vec{q}_{i-1,j,k} + A_{i,j,k}^p \vec{q}_{i,j,k} \\ + A_{i,j,k}^e \vec{q}_{i+1,j,k} + A_{i,j,k}^n \vec{q}_{i,j+1,k} + A_{i,j,k}^t \vec{q}_{i,j+1,k} = \vec{b}_{i,j,k} \end{aligned} \quad (2.37)$$

The coefficients A^b , A^s etc. are 4×4 matrices, the vector \vec{q} is the vector of primitive variables (u_{3i}, p) and \vec{b} the vector of right hand side (RHS) quantities. All the algebraic equations are put in this form, retaining the terms fitting the seven point difference molecule on the left hand side, and lumping the rest of the terms with the RHS vector \vec{b} . These equations can be put in the form of a matrix as:

$$[A] \vec{q} = \vec{b} \quad (2.38)$$

The matrix $[A]$ is of the following form:

$$|\bar{q}^{n+1,k+1} - \bar{q}^{n+1,k}| \leq \epsilon \quad (2.41)$$

When this convergence is satisfied, it can be seen from Equation 2.40 that the effect of the added matrix $[P]$ is removed. The solution procedure is best explained by putting these equations in the so called 'delta form' which is explained below. Denoting the incremental change in the primitive variable vector from one iteration to the next as the 'delta vector' as:

$$\vec{\delta}^{n+1,k+1} = \bar{q}^{n+1,k+1} - \bar{q}^{n+1,k} \quad (2.42)$$

Subtracting $[A + P]\bar{q}^{n+1,k}$ from both sides of Equation 2.40 gives:

$$[A + P]\bar{q}^{n+1,k+1} - [A + P]\bar{q}^{n+1,k} = [P]\bar{q}^{n+1,k} + \vec{b} - [A + P]\bar{q}^{n+1,k+1} \quad (2.43)$$

which can be rewritten as:

$$[A + P]\vec{\delta}^{n+1,k+1} = \vec{b} - [A]\bar{q}^{n+1,k} \quad (2.44)$$

Replacing the matrix $[A + P]$ with the appropriate LU product gives:

$$[L][U]\vec{\delta}^{n+1,k+1} = \vec{b} - [A]\bar{q}^{n+1,k} \quad (2.45)$$

At the end of every iteration, the right hand side of Equation 2.45 can be calculated. One sweep of forward substitution followed by a sweep of backward substitution yields the vector $\vec{\delta}^{n+1,k+1}$ required to update the primitive variables. This update continues until convergence at the present time level.

There are two important points to note here. These are:

1. *Linearization error removal*

It should be remembered that the nonlinear terms in the governing equations 2.32 were linearized using a Newton linearization procedure. To adequately resolve the non-linear aspects, the matrix $[A]$ (which is a function of the various primitive variable quantities besides other entities) need to be properly recomputed after every iterative update of the primitive variable vector.

2. *Elements of the $[L]$ and $[U]$ matrices*

Though it was mentioned that the matrix $[A + P]$ is suitable for expression as an LU product, the steps involved in this process have not been discussed yet. This is done through a set of predetermined operations on the elements of the matrix $[A]$. The discussion of these steps and the vectorization aspects of this calculation procedure is given in the following section.

2.6 Vectorization of the CSIP method

The sequence of steps involved in a typical liquid sloshing calculation using the surface method 'A' was given in the previous section. The subiteration at each time level involves one iteration of the CSIP solution where the increments to the primitive variable vector are estimated. These increments are then added to the solution vector and the process continues until convergence. It is clear that a significant fraction of the total computation time would be spent in this step (CSIP iterations). Hence, any savings in computational time associated with this procedure could result in significantly enhanced performance.

2.6.1 Vector processing

In contrast to the workstations mentioned earlier, some computers like the Cray Y-MP possess the ability to perform the same instruction on more than one data item simultaneously. This is defined as *vector processing* and the steps involved in modifying a calculation procedure to utilize this capability constitute vectorization.

The following is a quotation from the first page of the Cray User manual:

The difference in speed between good and poor code can be a factor of ten or more. The fundamental way in which the Cray supercomputers derive their speed is through vector processing. Vectorized code is approximately ten times faster, and can be up to 40 times faster, than scalar code.

It is evident that vectorization is essential for calculations using such computing architectures. The best way to explain vectorization would be by way of examples. An operation sequence that consists of multiplying all the elements of an array by the same constant is vectorizable; so is a loop that multiplies corresponding elements from two arrays.

Examples (in FORTRAN) of such operations are given below.

```
DO 100 I = 1,100  
A(I) = A(I)*5  
100 CONTINUE
```

and

```
DO 200 I = 1,100  
A(I) = B(I)*C(I)  
200 CONTINUE
```

DO loops in which one pass through the sequence requires results from a previous sweep are said to have *vector dependencies* or *recursions*. For example , a loop such as:

```
DO 300 I = 2,100  
A(I) = A(I-1)  
300 CONTINUE
```

has vector dependency.

The first two loops possess no vector dependencies and hence are vectorizable. That is, all elements of the array A could be multiplied by the constant '5' simultaneously; similarly all elements of the array B can be multiplied by corresponding element from array C and the result stored in A. All these could be done simultaneously.

The last loop simply transfers the value of A(1) to all array elements from A(2) through A(100). Vectorizing this loop would copy all the elements from A(1) through A(99) in one register and reassign these values starting at A(2) through A(100). Clearly, the scalar and vector processes give different results. If the output from the conventional scalar operation was in fact the desired result, nothing short of rewriting the algorithm would make this step vectorizable.

Though there exist many more nuances to vector operations, the ability to recognize this type of simple recursion is sufficient to follow the ensuing discussion on

$$h_{i,j,k} = g_{i,j,k}^{-1}(A_{i,j,k}^e - \alpha_2 \phi_{i,j,k}^2 - \alpha_1 \phi_{i,j,k}^3)$$

$$r_{i,j,k} = g_{i,j,k}^{-1}(A_{i,j,k}^n - \alpha_3 \phi_{i,j,k}^2 - \alpha_1 \phi_{i,j,k}^4)$$

$$v_{i,j,k} = g_{i,j,k}^{-1}(A_{i,j,k}^t - \alpha_2 \phi_{i,j,k}^6 - \alpha_3 \phi_{i,j,k}^5)$$

where α_1 , α_2 and α_3 are the partial cancellation parameter in the x , y and z directions respectively.

The definition of $\phi_{i,j,k}^1$, $\phi_{i,j,k}^2$, $\phi_{i,j,k}^3$, $\phi_{i,j,k}^4$, $\phi_{i,j,k}^5$ and $\phi_{i,j,k}^6$ are expressed as follows:

$$\phi_{i,j,k}^1 = a_{i,j,k} h_{i,j,k-1}$$

$$\phi_{i,j,k}^2 = a_{i,j,k} r_{i,j,k-1}$$

$$\phi_{i,j,k}^3 = d_{i,j,k} h_{i,j-1,k}$$

$$\phi_{i,j,k}^4 = f_{i,j,k} r_{i-1,j,k}$$

$$\phi_{i,j,k}^5 = d_{i,j,k} v_{i,j-1,k}$$

$$\phi_{i,j,k}^6 = f_{i,j,k} v_{i-1,j,k}$$

These equations were originally derived for scalar systems, i.e., there exists only one unknown at each grid location (i, j, k) . When the system is coupled as in the present case, the extension is fairly straightforward. Each element is an $n \times n$ matrix where n is the number of unknown variables at each grid location (4 in this problem). All multiplications are matrix-matrix multiplications and a division represents multiplication by the inverse of an $n \times n$ matrix.

2.6.3 Planes of constant index sums

The above sequence of operations is not best suited to utilize the vector capability of supercomputers if they are performed in the conventional order of doing such computations. The usual strategy would be to hold two of the three spatial indices at one level and increment the third index. This option eliminates the possibility of simultaneous operation on multiple data sets; in other words, this prohibits vectorization. However, the vector processing is possible by suitably rearranging the sequence in which the operations are performed. To achieve this, one first step would be to convert the three-dimensional domain into a two dimensional domain of planes containing points whose indices add up to the same value. A picture of such planes in a three-dimensional space is shown in Figure 2.5.

A look at the CSIP operations listed above would reveal that the sums of indices of any of the parameters on the right hand side of these equations is at least one less than the sum of the parameter on the left hand side. Hence when the three dimensional space has been converted into a two-dimensional space of planes containing points with constant index sums, any operation for any point on plane 'n' would not require parameters at any other point on plane 'n'. Besides, all the operations for any

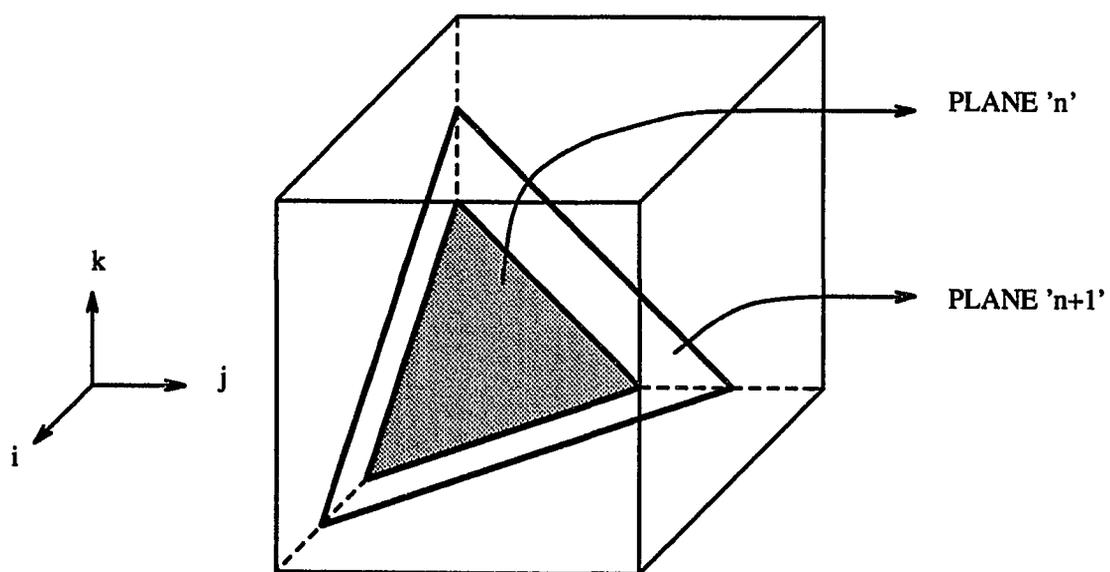


Figure 2.5: Planes of constant index sums in three dimensional space

point on plane 'n' would require information from planes with their index sums less than the index sum for the current plane. Hence by sweeping the calculations from the plane with the lowest index sum to the one with the highest sum, the operations are vectorizable; thereby the steps for all the points in a specific plane could all be done at once instead of one point after another.

Another important aspect of the vectorization deals with the inversion of the $n \times n$ matrices. On vector processors, many such matrices could be inverted at the same time; inversions involve operations on matrix elements and inversion of one matrix is independent of the other. These matrices should all be inverted in a vector fashion for the complete benefits of the vectorization to be realized.

In all the calculations discussed in this thesis, the vectorized version of the CSIP algorithm has been used. A listing of the FORTRAN subroutines involved in the CSIP method is given in Appendix B. The speed-up obtained due to the vectorization has been tremendous. A megaflop rating of 132 compared to a mere 4 Mflops for the scalar version was observed on the Cray Y-MP. The relevant portions of output from a performance analysis tool available on the Cray series supercomputers is given in Appendix C.

2.7 Smoothing

Oscillations in the calculated solutions to incompressible Navier-Stokes equations have been observed in previous studies. These have been attributed to the use of central differences in the spatial discretization of the governing equations. Traditionally, two approaches have been followed to overcome this phenomenon: upwinding and smoothing.

As previously mentioned, central differences have been used to approximate partial derivatives in all the spatial directions. It was noticed that the solutions of primitive variables obtained had some oscillations in one or more spatial directions. These spurious profiles were found along radial directions for calculations of liquid sloshing in spherical and cylindrical containers and along both streamwise and transverse directions in the broken dam computations. Use of upwind differences for the convective terms did not eliminate this behavior. Hence a fourth-order smoothing of the following type was used for all the primitive variables after convergence at every time-step.

$$q_i = q_i - \omega [q_{i+2} - 4q_{i+1} + 6q_i - 4q_{i-1} + q_{i-2}]$$

A smoothing constant of $\omega_s = 0.05$ was generally used for the calculations discussed in this thesis. Smoothing, in general, was not required for the liquid sloshing calculations performed by Chen. He observed that smoothing the pressure alone was found to be sufficient and rarely was the smoothing required for all the variables. But, the computations performed in this research are at significantly higher Reynolds numbers compared to Chen's calculations [17]. The requirement for the enhanced smoothing could be attributed to these higher speeds.

3. LIQUID SLOSHING INSIDE SPHERICAL CONTAINERS

One version of the surface-fitting calculation procedure using the surface method 'A' has been employed by Chen [17] [18] to calculate a variety of axisymmetric as well as three-dimensional sloshing motion of liquids inside partially filled spherical containers. These calculations included the start-up from different initial conditions, namely initially capped, as well as impulsively started container motions. In a later effort, the motion of a gradually started orbital spin-up of a sphere, half filled with glycerine has also been reported [51]. This paper includes comparison of the transient results with experimental data obtained from the satellite test rig at Iowa State University. The comparison was found to be reasonably good.

However, in one calculation of a three-dimensional sloshing motion of a sphere that was partially filled with glycerine, some problems were encountered. These problems were generally more noticeable in calculations with relatively more surface deformation, and relatively higher container velocities than cases with lower speeds. These problems were two-fold:

1. *Global mass conservation*

The calculation scheme assumed a closed container and hence no liquid leaves the spherical tank. During the calculations, it was noticed that the difference between the volume (or mass, as the fluid has constant density) of liquid initially

present in the tank and the values calculated at the end of every time-step, differed significantly. The difference between these two quantities was found to be approximately 10% of the initial volume contained in the tank.

2. Pseudo-time step convergence and overall stability of the calculations

Another interesting aspect that was observed, typically more prominently at higher tank rotational speeds, was that the number of subiterations at each physical time step kept growing without bound. This phenomenon, in turn caused the calculations to become unstable and eventually blow up.

The algorithm and the resultant computer code were developed by Chen to be run on the workstations (Apollo DN 3500 and Apollo DN 10,000 and DECstation 500/200) available on campus at that time. It was believed that choosing a finer grid (compared to the $41 \times 11 \times 11$ used generally for most calculations) could at least alleviate, if not completely eliminate, some of the above problems. This could not be done owing to the memory limitations on the computer resources available. Besides, the finer grid calculations generally tended to be prohibitively time consuming.

Through a combination of modifications to the boundary condition implementation procedures and optimization of the computer code these problems have been overcome in the present research. This chapter deals with the many modifications and improvements done to the surface fitting algorithm since its initial implementation. Some of the topics discussed include:

1. Effects of vectorization

2. Code optimization

3. Contact line boundary conditions

4. Improved free-surface calculation procedure

Results for the orbital spin-up of a spherical container half-filled with glycerine at moderate Reynolds numbers, calculated with all these modifications, are also presented.

3.1 Effects of vectorization

The vectorization of the CSIP algorithm has been explained in detail in the last chapter. Besides this routine, many other parts of the computer code were identified for vectorization. Various measures were taken to vectorize different parts of the code. Some of the important ones included eliminating recursions by changing the strides in DO loops and also expanding subroutines to include some vector loops. An example of a subroutine expansion to include a loop is given below. The non-vector form is given first, followed by the equivalent but vector form of the code.

```
DO 100 I = IMIN,IMAX  
CALL EXAMPLE(X,Y,Z,I)  
100 CONTINUE
```

```
CALL EXAMPLE(X,Y,Z,IMIN,IMAX)
```

```
.  
. .  
. . .
```

```
.  
.   
SUBROUTINE EXAMPLE(X,Y,Z,IBEGIN,IEND)
```

```
DO 100 I = IBEGIN,IEND .
```

```
.  
.   
100 CONTINUE
```

In all the calculations discussed in this thesis, the vectorized version of the CSIP algorithm has been used. A listing of the FORTRAN subroutines involved in the CSIP method is given in Appendix B. The speed-up obtained due to the vectorization has been tremendous. A megaflop rating of 132 compared to a mere 4 Mflops for the scalar version was observed on the Cray Y-MP. The relevant portions of output from a performance analysis tool available on the Cray series supercomputers is given in Appendix C.

3.2 Code optimization

Besides the vectorization of the CSIP portion of the computations, it was possible to make the numerical calculations computationally more efficient through other means. As stated earlier, the procedure and the computer code was initially tailored to run on campus workstations. Wherever possible, storage of data was avoided to fit the memory capacity of these smaller machines. In such cases, the same numbers (floating point data sets forming arrays) were calculated again and again instead

of estimating them just once and storing their values. This resulted in a reduced memory requirement but the price was slower progress of the numerical calculations.

For the calculations performed in this research, this procedure could be changed due largely to the much higher memory capacity of the supercomputers. A key part of the calculations where this change resulted in significant computational savings relates to the non-linear convection terms in the momentum equations. The final form of the momentum equations governing the liquid sloshing calculations are of the form:

$$\begin{aligned} \frac{\partial u_{3n}}{\partial \tau} + (\dot{z}_j + \eta_{j,i} u_{3i} + \eta_{j,i} f_{ik} x_{3k}) \frac{\partial u_{3n}}{\partial z_j} - (f_{ni} + 2\lambda_{ni}) u_{3i} \\ - \eta_{j,n} \frac{\partial p}{\partial z_j} + \frac{1}{Re} (\eta_{j,i} \eta_{k,i} \frac{\partial^2 u_{3n}}{\partial z_j \partial z_k} + \eta_{k,ii} \frac{\partial u_{3n}}{\partial z_k}) \\ = -\tau_{1,ni} x_{3i} - \tau_{2,ni} h_i - g_{3n} + E_n, \end{aligned} \quad (3.1)$$

The nonlinear term was linearized using a Newton linearization procedure [50]. For example, terms of the type $u_{3i} \frac{\partial u_{3n}}{\partial z_j}$ were approximated as:

$$\begin{aligned} \left[u_{3i} \frac{\partial u_{3n}}{\partial z_j} \right]^{n+1,k+1} \simeq (u_{3i})^{n+1,k} \left[\frac{\partial u_{3n}}{\partial z_j} \right]^{n+1,k+1} \\ + \left[\frac{\partial u_{3n}}{\partial z_j} \right]^{n+1,k} (u_{3i})^{n+1,k+1} - \left[u_{3i} \frac{\partial u_{3n}}{\partial z_j} \right]^{n+1,k} \end{aligned} \quad (3.2)$$

where 'n' represents time level and 'k' stands for subiteration count.

Once this was done, terms that fit into the CSIP seven point molecule were retained on the left hand side (as part of the matrix [A]) and the rest of the terms were lumped to the RHS. The matrix [A] had to be updated after every iteration of the CSIP routine. This was required as any update of the velocity components

necessitates a reevaluation of the matrix as they form part of the linearized convection term. Initially, the procedure adopted was to reset all the elements of the $[A]$ matrix to zero and then calculate each term and add them to the appropriate coefficients. Careful look at the momentum equations suggests that this is unnecessary; only the nonlinear convection terms have the primitive variables within the coefficients of the matrix $[A]$. Other terms do not involve the primitive variables as coefficients multiplying other primitive variables.

Hence the procedure was revised so that the matrix $[A]$ is split into two parts, one fixed part and the other varying:

$$[A] = [A]_{fixed} + [A]_{varying}$$

The varying part involves the first two terms of the linearized convection term that forms the RHS of Equation 3.3. Only these terms cause any change to the matrix $[A]$ from one subiteration (k) to the next ($k+1$). All the others terms from the governing equations that fit into the seven point molecule and hence fall into the LHS, form the fixed part of matrix $[A]$. Only the varying part of the matrix $[A]$ was evaluated after every velocity component update (which occurs once every subiteration). The fixed part was evaluated only once prior to the first pseudotime subiteration in every time step.

Although it appears that this change doubles the memory requirement compared to storing just the matrix as one whole unit instead of two subsets, it is not exactly true. In the varying part, the nonzero components fall along the diagonals of the corresponding 4×4 matrices. All the off-diagonal elements inside each of these matrices are zero in the varying component of the larger matrix $[A]$. Hence the

Table 3.1: Performance comparison of scalar and vector computer codes

Item	Scalar code	Vectorized code	Factor of speed-up
Megaflop rating			
CSIP routine	4	132	33
whole program	3	62	20
CPU time consumed, seconds			
whole program	154	13	12

memory requirement increases only by 25% of the requirement for [A] compared to the previous method.

This modification resulted in significant computational savings. Besides this, other changes that could potentially increase the computational efficiency were identified and implemented. One common aspect among these changes was that repetitive calculation of the same constants during every subiteration was replaced by calculating them just during the first subiteration and storing them for later use.

The vectorization and code optimization explained so far produced an enormous increase in the computational speed of the calculations. Table 3.1 gives the comparison of execution rating and time consumed by the calculations between the original version of Chen and the vectorized version.

3.3 Contact line boundary condition

The curve where the free-surface intersects the solid wall of the container is referred to as the contact line. A singularity exists in the vicinity of this contact line;

the singularity is stronger in the case of dynamic (or moving) contact lines compared to stationary (or static) contact lines.

The singularity can be viewed as the result of the combination of having to use the no slip boundary conditions on fluid elements in contact with the container wall just below the free surface and at the same time requiring the fluid surface in contact with the wall to move from one time step to the next. Due to this, the stress becomes infinite near the contact line [43]; the extent of difficulty in treating the singularity depends on whether the contact line is static or moving [44].

The moving contact line has been studied from the point of the nature of the physical conditions associated with the problem [45] [46] [47]. Though it is generally suggested that slip boundary conditions show promise in the case of dynamic moving lines, no specific boundary condition has been suggested as being superior to others for the case of liquid sloshing in containers. But the general suggestion has been that the right kind of boundary conditions to be used on the contact line is problem dependent. Utility of various types of slip conditions have been discussed in the literature [48] [49].

For all the cases discussed in this thesis, a condition that the free-surface be normal to the wall at the point of contact has been used. Mathematically, this condition states that $\frac{\partial F}{\partial n} = 0$ at the contact line. This condition has been found to give the best results for the liquid sloshing computations presented in this study.

There were two distinct advantages to this boundary condition. The first of these was a better interior grid which is generated algebraically. In the grid generation scheme used for the sloshing problems, the grid lines inside the liquid were generated in conformance with the free-surface shape. The zero normal slope condition results in

normal or near-normal lines at the wall. Options such as extrapolating for the contact point from interior free-surface profiles, as well as other slip conditions resulted in very small angles between the free-surface and the wall. These acute angles in turn resulted in decreased stability and eventual blowing up of the calculations.

The second benefit of this condition was that the transient results were in better agreement with experimental data, especially in the case of low Reynolds number or high viscosity fluid flows. It should however be pointed out that the transient motion of the contact line was not sensitive to the condition used in the case of high Reynolds number flows. Better interior grid and resultant enhanced stability was however observed in all cases, of both high and low viscosity fluids.

The satellite test rig, its free-surface instrumentation, and some of the experiments have been described in an earlier paper [51]. A typical comparison with experimental data for the inboard end free-surface contact point motion in the case of two different contact boundary conditions is shown in Fig 3.1. These results are for the gradual spin-up from rest of a spherical tank half filled with glycerine to a final rotational velocity of 60 rpm. The spheres are mounted on an arm whose length is approximately 3.4 times the radius of the sphere.

3.4 Free-surface motion: surface method 'B'

The free-surface motion is computed by solving an equation based on the kinematic condition which simply states that any particle on the free-surface stays on the free-surface. For the case of sloshing in spherical containers, this equation is :

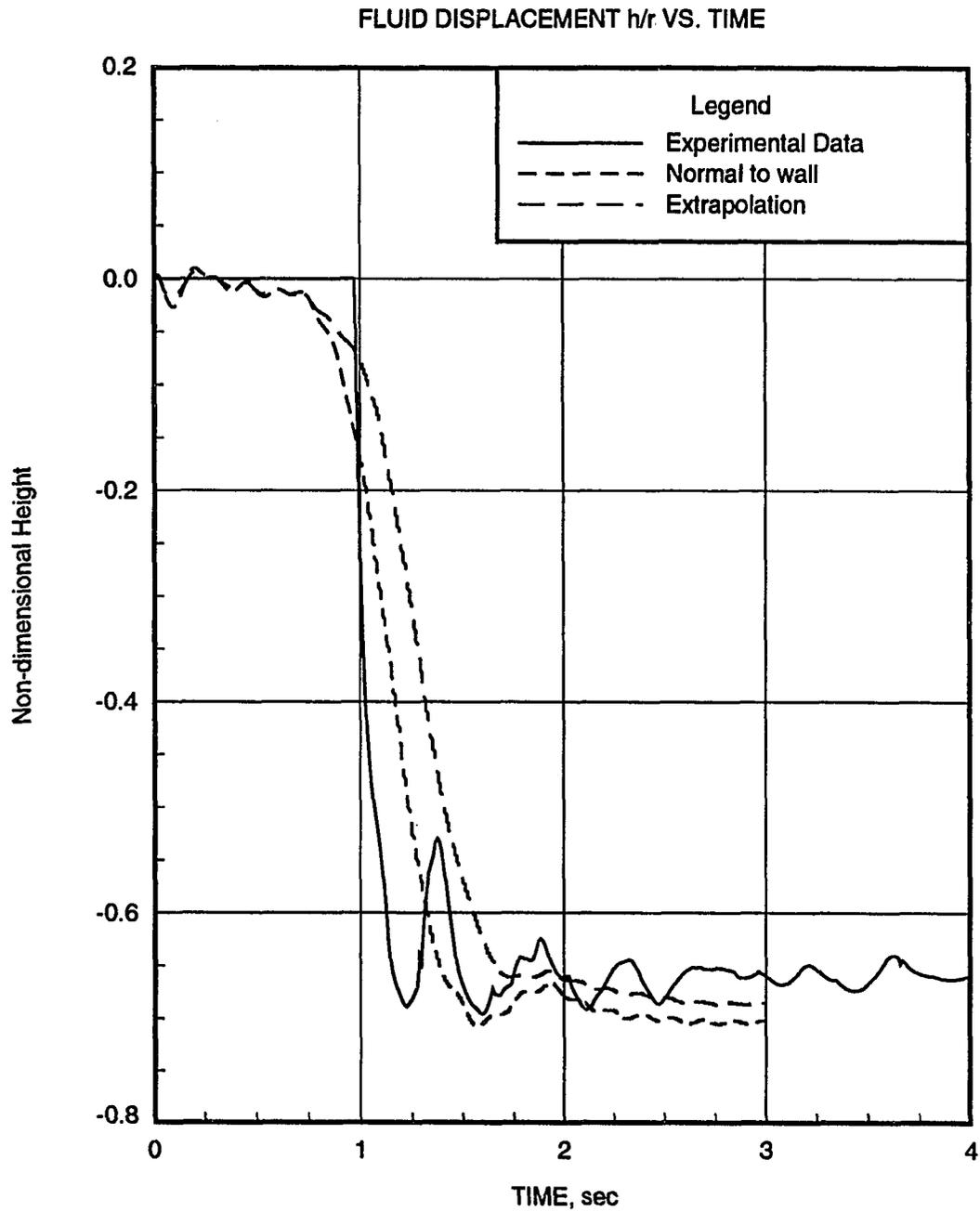


Figure 3.1: Comparison of numerically computed in-board end free-surface transience and experimental data for different contact line boundary conditions

$$\begin{aligned}
\frac{\partial F}{\partial \tau} = & \{u_{33} + f_{3k}x_{3k}\} \\
& - \{\dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2}\} \frac{\partial F}{\partial z_1} \\
& - \{\dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2}\} \frac{\partial F}{\partial z_2} \quad (3.3)
\end{aligned}$$

where $F = F(\tau, x_{31}, x_{32})$, a function of two spatial variables and time. Originally this equation was solved at the end of every time-step by lagging all the flow variables (u_{3i}), grid speed terms (\dot{z}_i), and the metric coefficients ($\eta_{i,j}$). The free-surface variable was however treated implicitly using a backward difference in time and central differencing for the spatial derivatives.

The finite difference expression used was:

$$\begin{aligned}
F^{n+1} = & F^n + \Delta\tau \{u_{33} + f_{3k}x_{3k}\} \\
& - \{\dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2}\} \frac{\partial F}{\partial z_1} \\
& - \{\dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2}\} \frac{\partial F}{\partial z_2} \quad (3.4)
\end{aligned}$$

For some calculations, upwind differencing was used in the spatial discretization to suppress saw tooth like free-surface profiles.

It was noticed that this method of lagging all the unknown parameters to the previous time-step had difficulty in conserving the overall mass of liquid; the mass (volume) of liquid in the container after each time-step was steadily decreasing. Attention was focussed on the lagged terms in solving the free-surface kinematic equation and a new procedure was devised. This procedure will hereafter be referred to as "surface method 'B'". The steps comprising this method 'B' are:

1. Solve the kinematic equation for the free surface position using Eq. (3.4)
2. Generate a grid, compute grid speed, metric terms etc.
3. Compute elements of the matrix A, and vector 'b' in $Aq = b$
4. Calculate increments for the primitive variables by CSIP
5. Update the primitive variables
6. Check for convergence
7. IF converged,

(a) Solve for revised free-surface information by using the equation:

$$\begin{aligned}
 \left[\frac{\partial F}{\partial t} \right]^{n+\frac{1}{2}} &= \frac{F^{n+1} - F^n}{\Delta t} \\
 &= \frac{[C_1]^n \left[\frac{\partial F}{\partial z_1} \right]^n + [C_1]^{n+1} \left[\frac{\partial F}{\partial z_1} \right]^{n+1}}{2} \\
 &+ \frac{[C_2]^n \left[\frac{\partial F}{\partial z_2} \right]^n + [C_2]^{n+1} \left[\frac{\partial F}{\partial z_2} \right]^{n+1}}{2} + \frac{[S]^n + [S]^{n+1}}{2}
 \end{aligned} \tag{3.5}$$

where

$$C_1 = \{ \dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2} \}$$

$$C_2 = \{ \dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2} \}$$

and

$$S = \{ u_{33} + f_{3k}x_{3k} \}$$

- (b) Generate a grid, compute grid speed, metric terms etc.
- (c) Compute elements of the matrix A, and vector 'b' in $Aq = b$

- (d) Calculate increments for the primitive variables by CSIP
 - (e) Update the primitive variables
 - (f) Check for convergence
 - (g) IF converged,
 - i. If Eqn. 3.5 has been solved desired number of times, increment time level; go to step 1. If not, go to step 7 (a).
 - ii. ELSE, repeat steps 7(c) through 7(f) until time level can be incremented
- ELSE
8. Go to step 3 .
- ENDIF

The terms containing superscripts ' $n + 1$ ' were evaluated at the present time level at which other primitive variable solutions are sought. It can be seen from the expressions for the coefficients, C_1 and C_2 that they depend on velocities u_{3i} , grid speeds \dot{z}_i , and geometric parameters, $\eta_{i,j}$ and f_{ij} . Evaluation of these coefficients at the present time level requires such information at time level ' $n + 1$ '. Both equations 3.5 and 3.4 are finite difference forms of the kinematic condition given by Eqn. 3.3. However, it should be noted that the time-centered implicit differencing [50] in Eqn. (3.5) is closer to being second order accurate in time than the differencing used in Eq. (3.4). The higher-order method could not be implemented at step 1 of the above procedure as the information on the coefficients and the source term are not

available at time level (n+1). The necessary information is however available at step 7 (a) owing to the updates in steps 5 and 2.

3.5 Results for the calculation of the orbital spin-up of a spherical tank

This revised method drastically improved the global mass conservation behavior of the procedure. The steps 7 (a) through 7 (g) could be repeated for better results; one possible measure of when to stop this cycle is by requiring the number of times of passing through steps 7(c) through 7(f) to be smaller than a predetermined value. A value of unity was generally assigned. Convergence at step 7(g) within one iteration of steps 7(b) through 7(f) suggested that the changes to the primitive variables, and grid parameters were small and so would be the changes to coefficients in Eqn. 3.5. This could in turn be interpreted as a reasonably accurate solution for F .

Calculations of an orbital spin-up of a spherical tank half-filled with glycerine have been performed with repeating steps 2 through 7 twice as well as four times. The initial condition for this calculation belongs to the 'initially capped' class; the fluid with a flat surface is assumed to have achieved solid body rotation at the prescribed rotational velocity of the tank. The cap holding the free-surface in place is removed and the surface allowed to move and eventually reach its steady state position. For this case, the rotational speed was 60 rpm and the rotational arm length was 3.4 times the radius of the sphere.

The variation of the error in mass, which is the difference between the mass of liquid initially in the tank and the mass computed at the end of each time-step, with time steps is shown in the Figure 3.2. The final error in the calculation that used four free-surface updates in every time-step, is around 0.5 % which is a reasonable

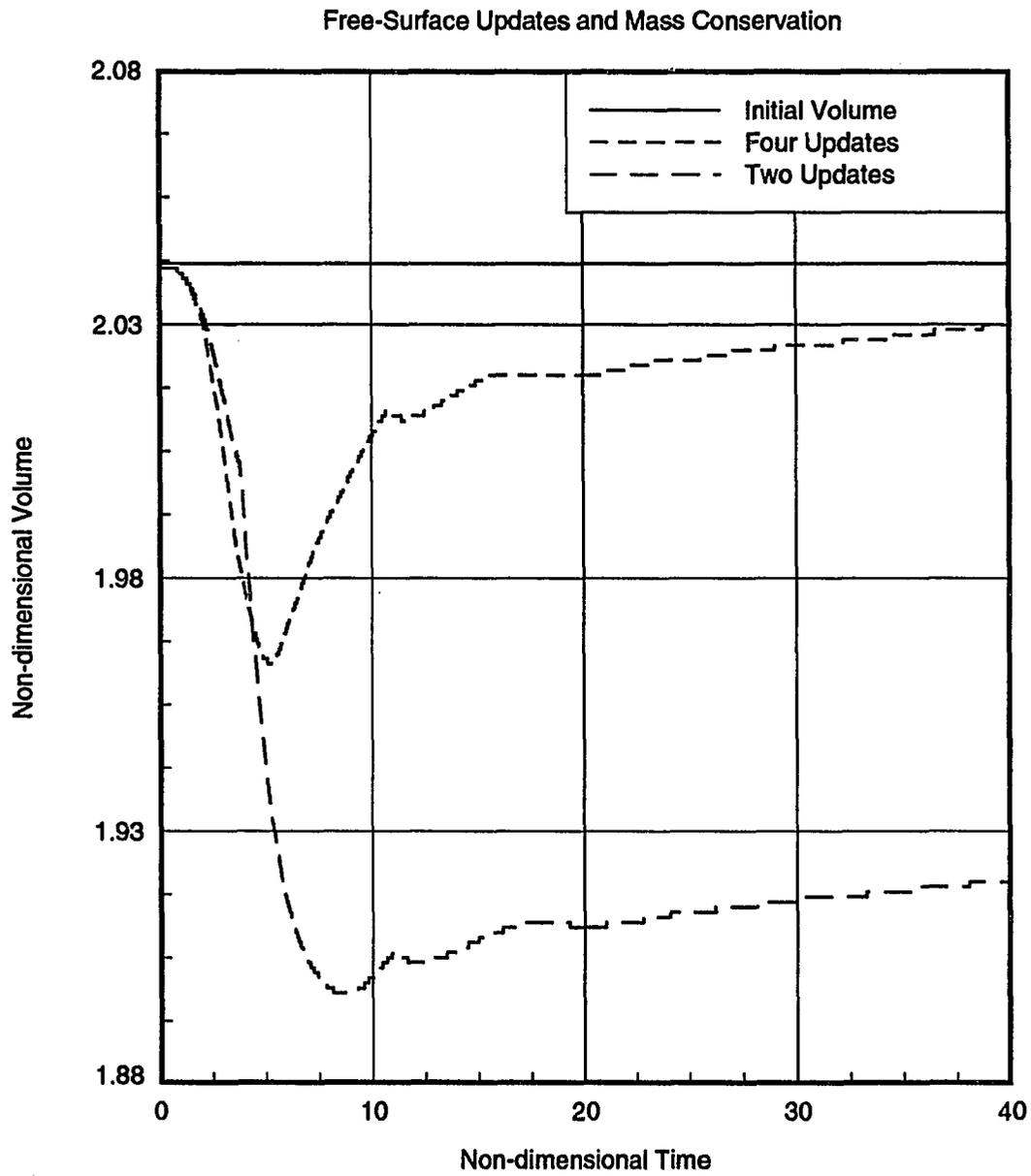


Figure 3.2: The effect of number of free-surface updates on the global mass conservation

value. Most of the higher errors found during the transient part of the solution can be attributed to the truncation errors in the integration scheme used to find the volume of each grid cell. The logic behind this reasoning follows from the fact that this error is higher when the free-surface is curved and the error reduces when the surface flattens as it approaches its steady state position. The sequence of free-surface profiles and velocity vector plots in the plane $x_{32} = 0$, at select instances are shown in Figures 3.3 and 3.4. The velocity vector plots do seem to be consistent with the free surface shape and the general flow pattern. For this case, the transient history of the motion of the contact points at the inboard and outboard ends are also shown, along with their analytical steady state values, in Figure 3.5. The computed final positions are in excellent agreement with the analytical results. The final steady state free surface profile in the $x_{32} = 0$ plane is shown for both the two and four update runs in Figure 3.6. It can be seen from this figure as well that the run using four free-surface updates in every time step compared better with the analytical solution. It should also be pointed out that the same case run earlier by Chen using the surface method 'A' compared even worse with the analytical results.

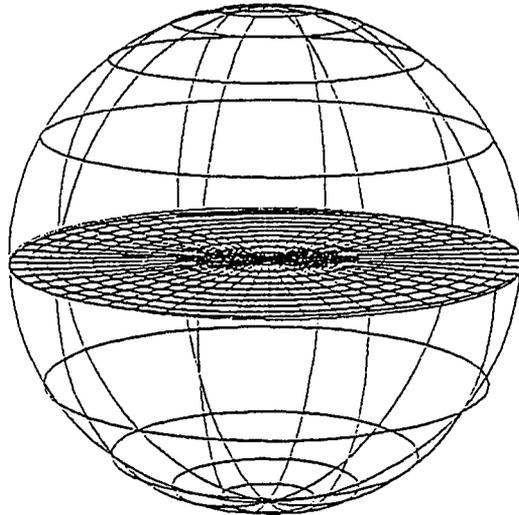
The nondimensional parameters for the above spherical container sloshing run were:

$$\text{Reynolds Number, } Re = 181.4$$

$$\text{Weber Number, } We = 12002.1$$

$$\text{Froude Number, } Fr = 3.25$$

In estimating the above dimensionless parameters, the radius of the sphere has been used reference length and the linear velocity of the tank center has been used as the reference velocity.



$$\tau = 0.1$$

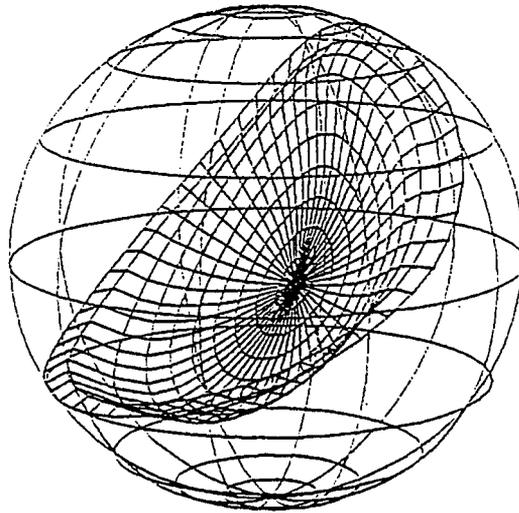


Figure 3.3: Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

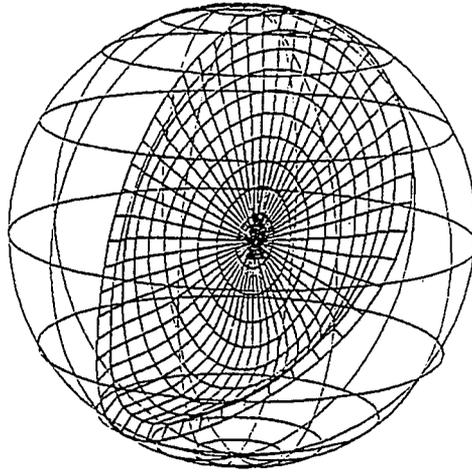
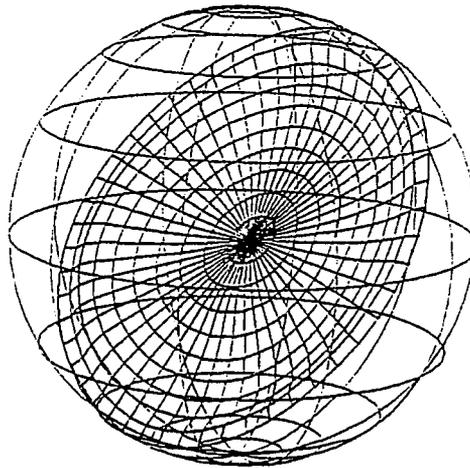
 $\tau = 8.0$  $\tau = 12.0$

Figure 3.3 (Continued)

Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

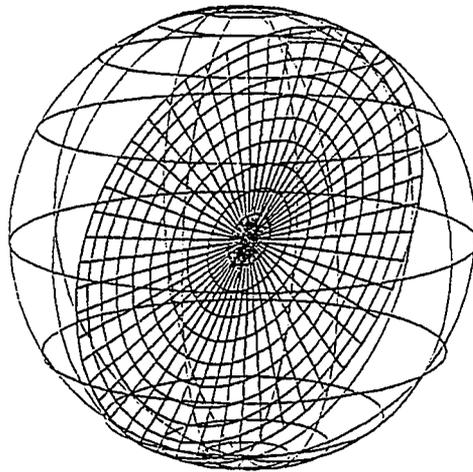
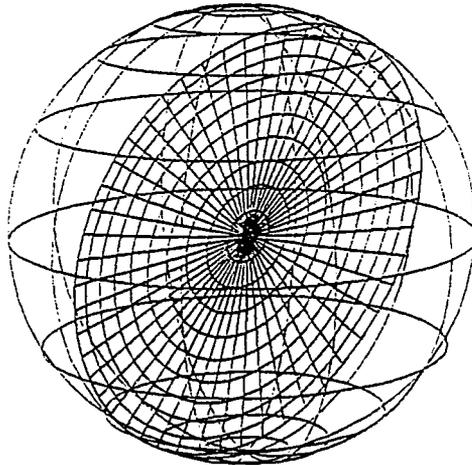
 $\tau = 32.0$  $\tau = 40.0$

Figure 3.3 (Continued)

Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

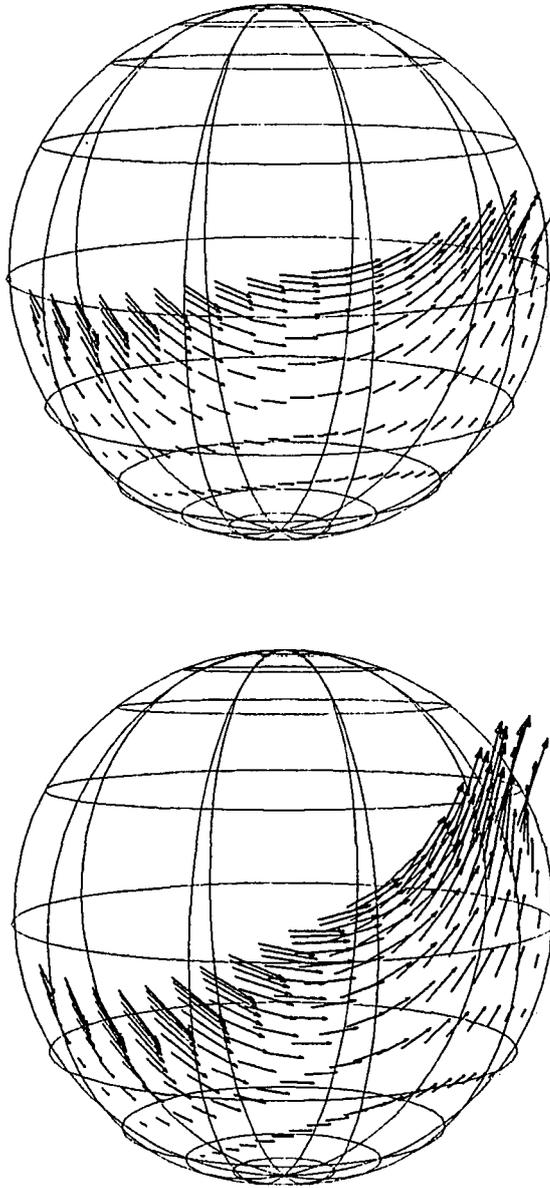


Figure 3.4: Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

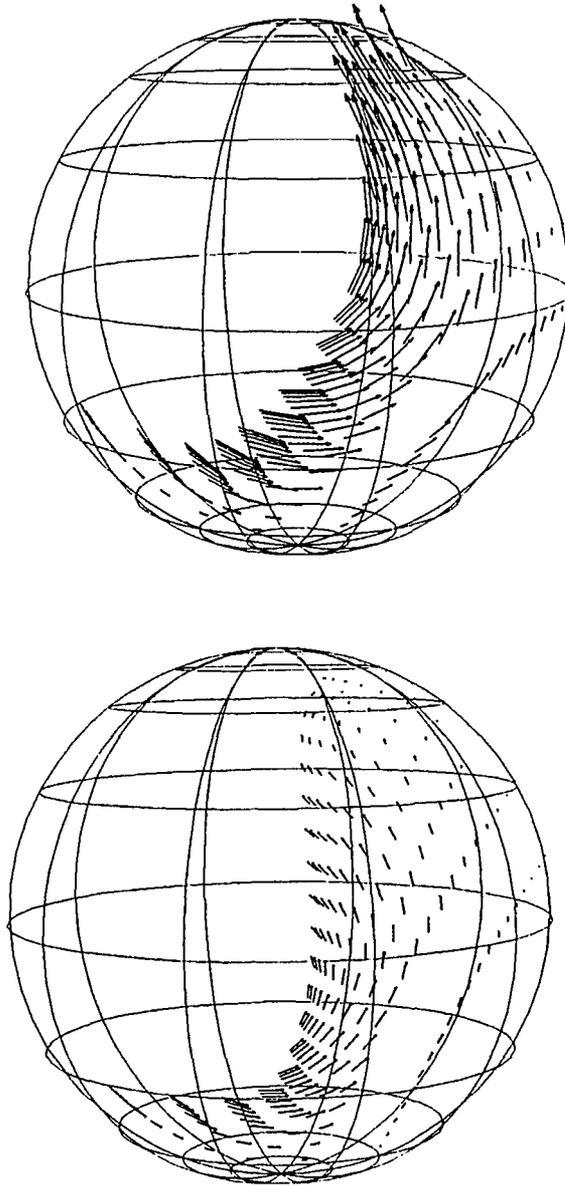


Figure 3.4 (Continued)

Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

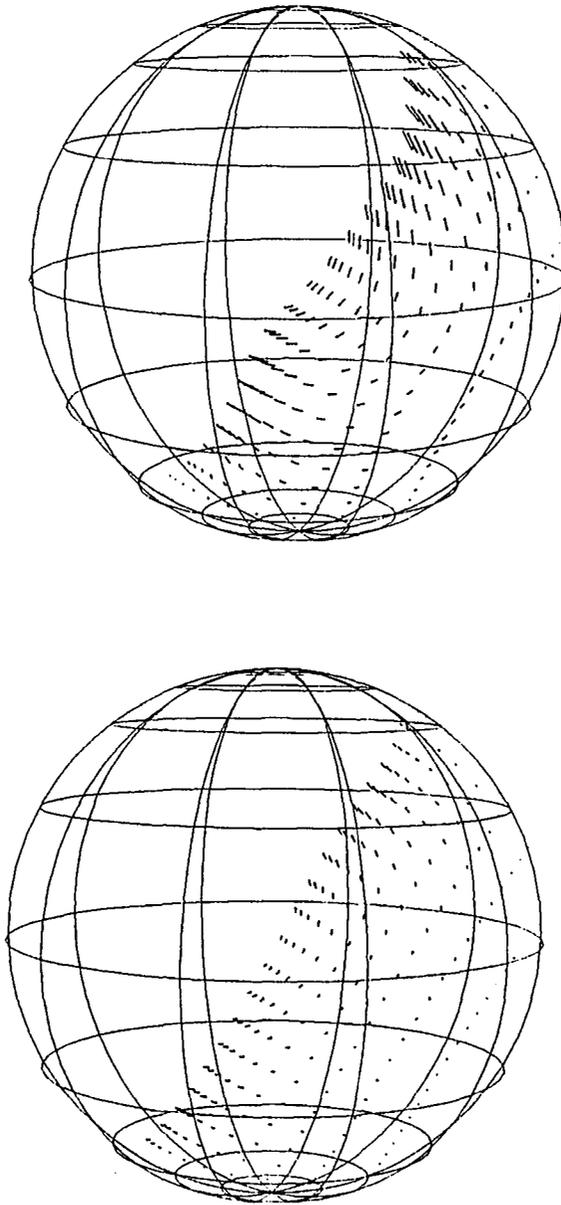


Figure 3.4 (Continued)

Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank

TRANSIENT MOTION OF CONTACT POINTS

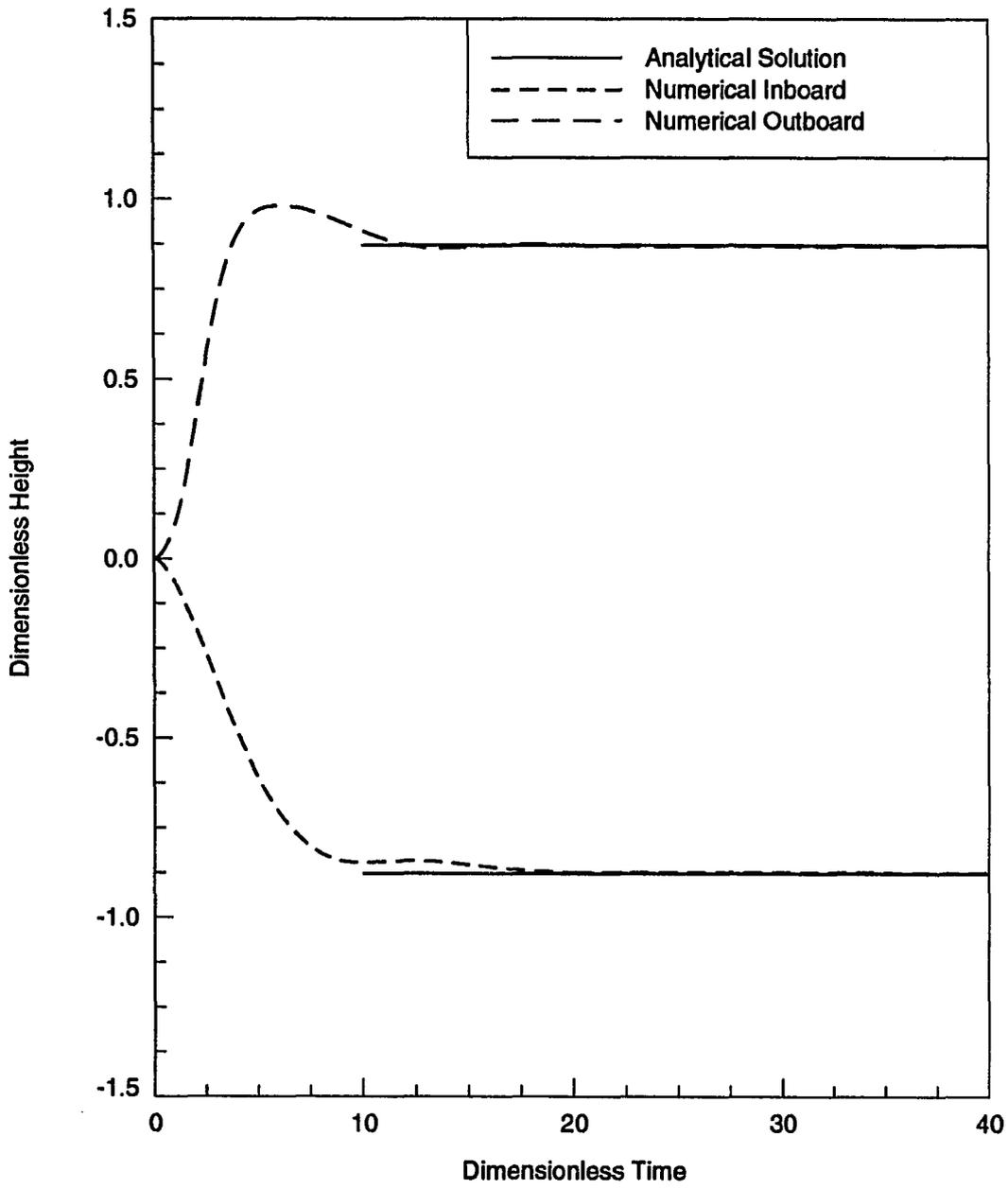


Figure 3.5: Transient motion of the inboard and outboard free-surface contact points, including analytical steady state values

STEADY STATE FREE SURFACE PROFILES

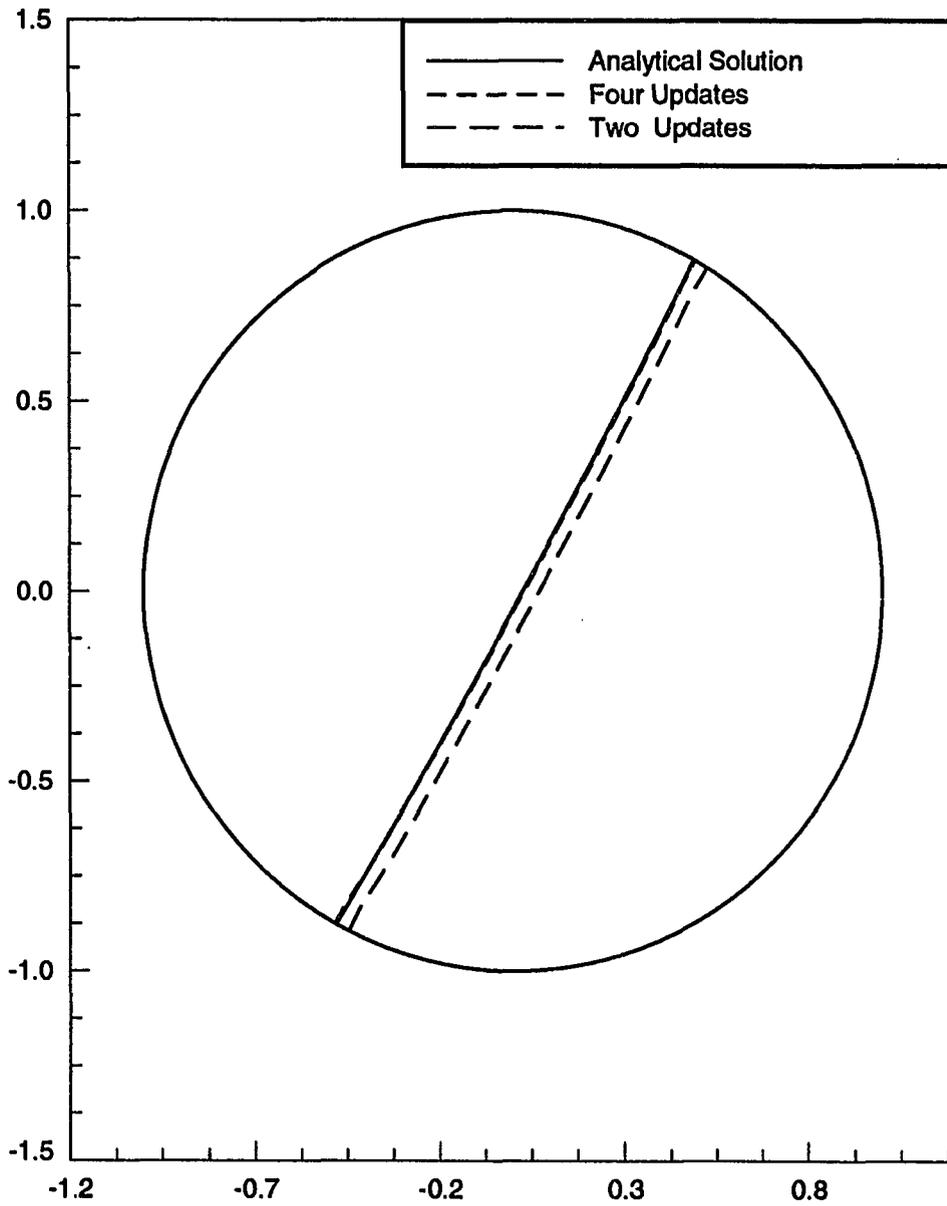


Figure 3.6: Comparison of the analytical and numerical steady state free-surface profile in the plane $x_{32} = 0$ for different number of free surface updates

Prior to the implementation of the modifications listed in this chapter, this case was the most challenging three-dimensional liquid sloshing calculation attempted using the surface fitting code. However, this attempt did not produce results that were fully satisfactory [17]. As a means of further verification of the new surface calculation procedure, a more severe case was attempted. All the parameters were kept the same as in the earlier run excepting the rotational arm length, which was increased to 9.7 times the radius of the sphere from the earlier value of 5.8. The starting conditions were the same initially capped conditions as before. The nondimensional numbers for this case were:

$$Re = 300.0$$

$$We = 32840.0$$

$$Fr = 5.373$$

The free surface profiles, and the velocity vector plots in the plane $x_{32} = 0$ at different transient instances for this run are shown in the Figures 3.7 and 3.8. The free-surface undergoes significantly more deformation in this case compared to the previous case. The transient motion of the inboard and outboard free-surface contact points are shown in Figure 3.9 and the comparison of the final free surface profile in the $x_{32} = 0$ plane with the analytical solution is shown in the next Figure 3.10. The outboard contact point reaches the top of the sphere (where the vertical axis meets the sphere) during the transient motion; this duration can be identified in Figure 3.9 as the period where the height (y axis value) equals unity. Excellent agreement between computed and analytical free-surface profiles can be observed in Figure 3.10.

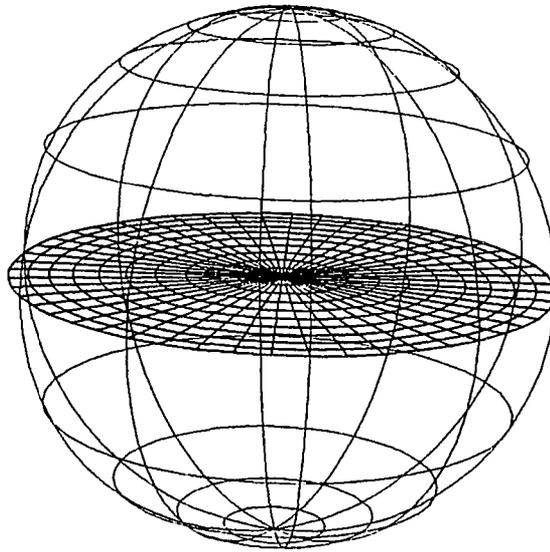
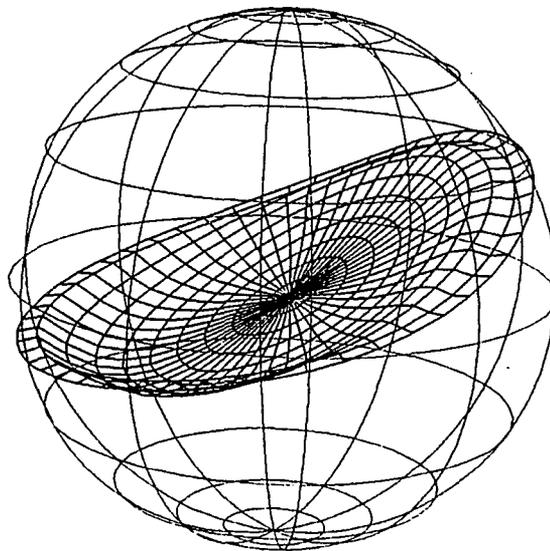
 $\tau = 0.00$  $\tau = 0.30$

Figure 3.7: Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$

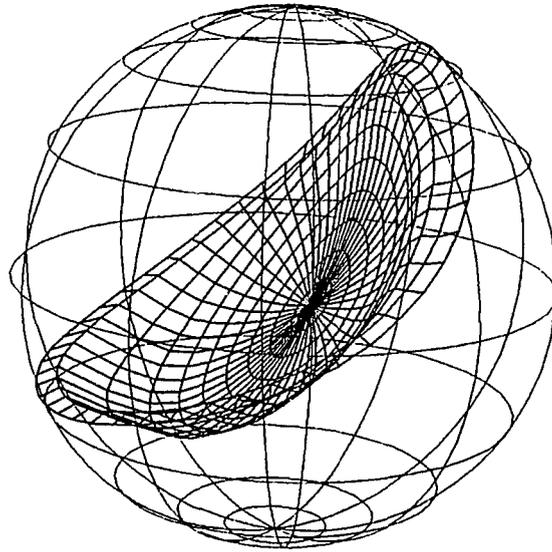
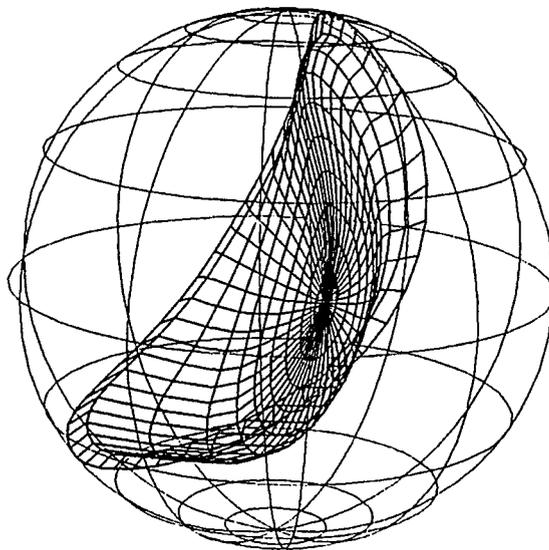
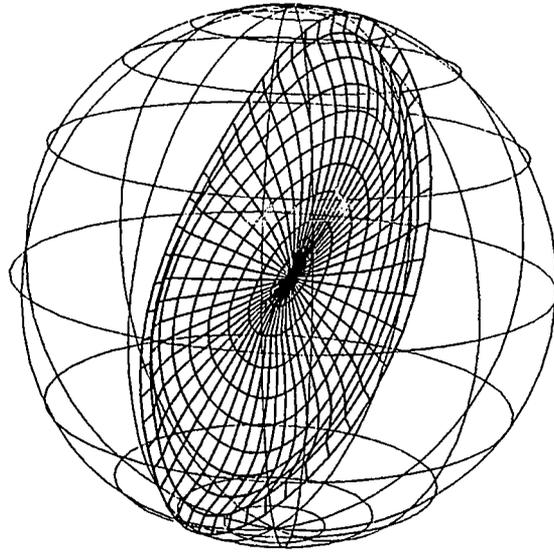
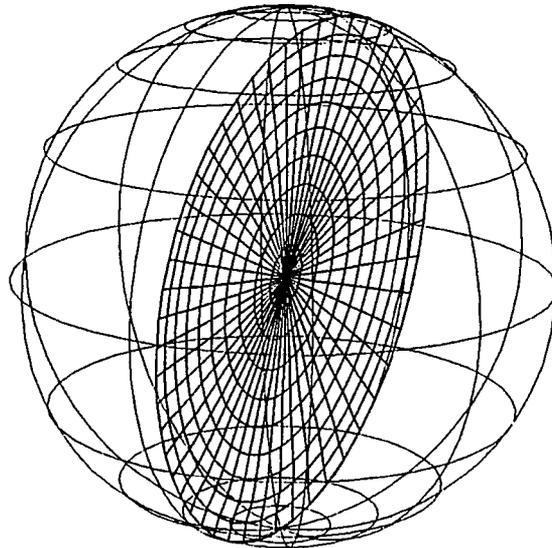
 $\tau = 0.45$  $\tau = 0.60$

Figure 3.7 (Continued)

Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$



$\tau = 1.50$



$\tau = 4.00$

Figure 3.7 (Continued)

Free-surface profiles at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$

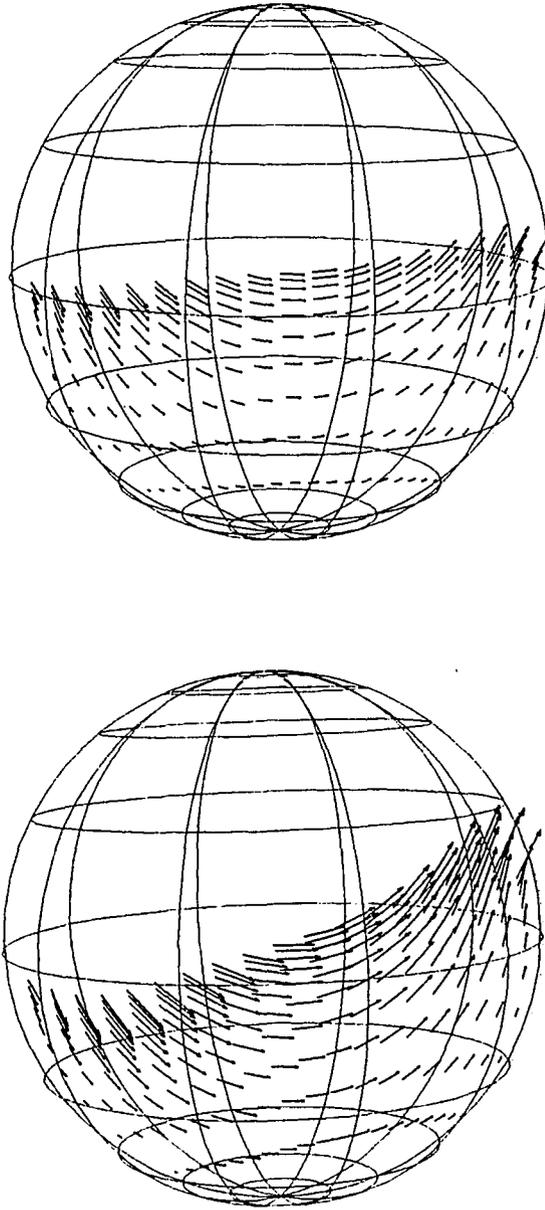


Figure 3.8: Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$

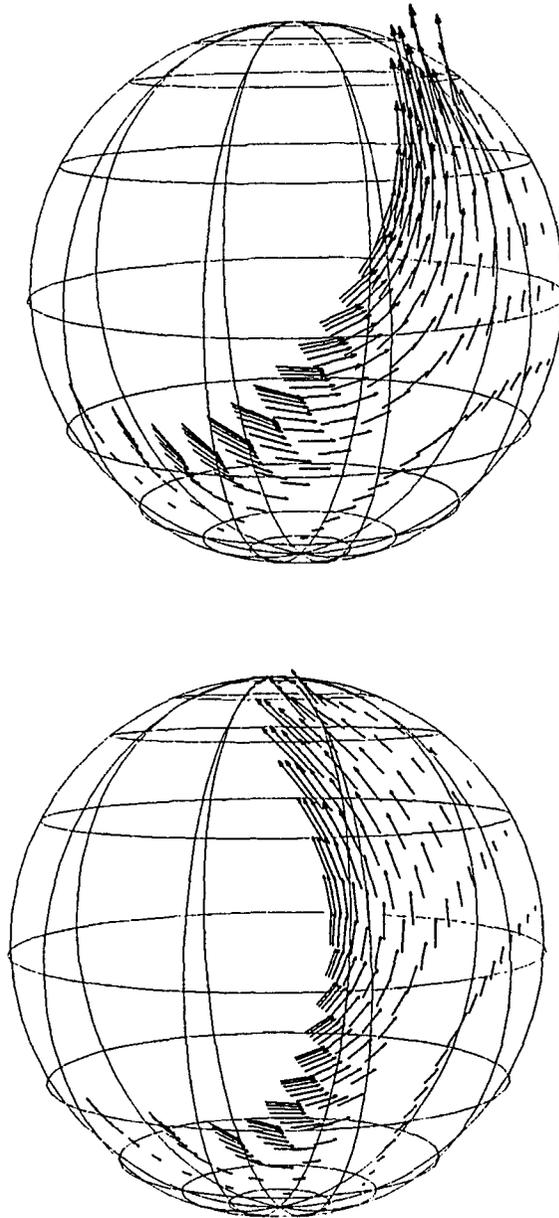


Figure 3.8 (Continued)

Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$

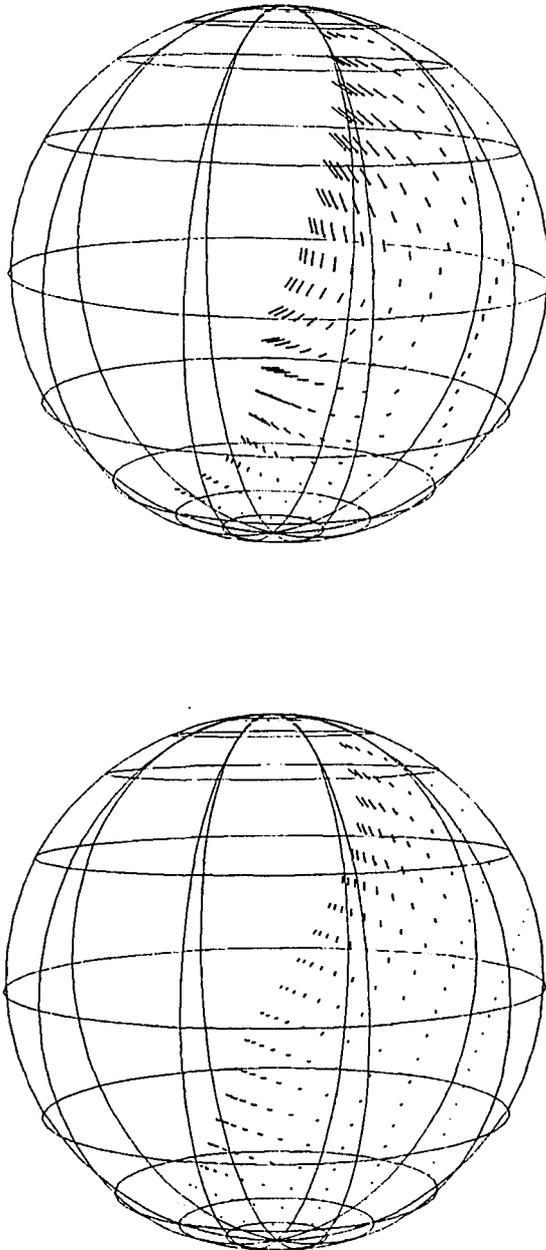


Figure 3.8 (Continued)

Velocity vector plots on the $x_{32} = 0$ plane at different instances during the asymmetric initially-capped spin-up of a half-filled spherical tank, $Re = 300$

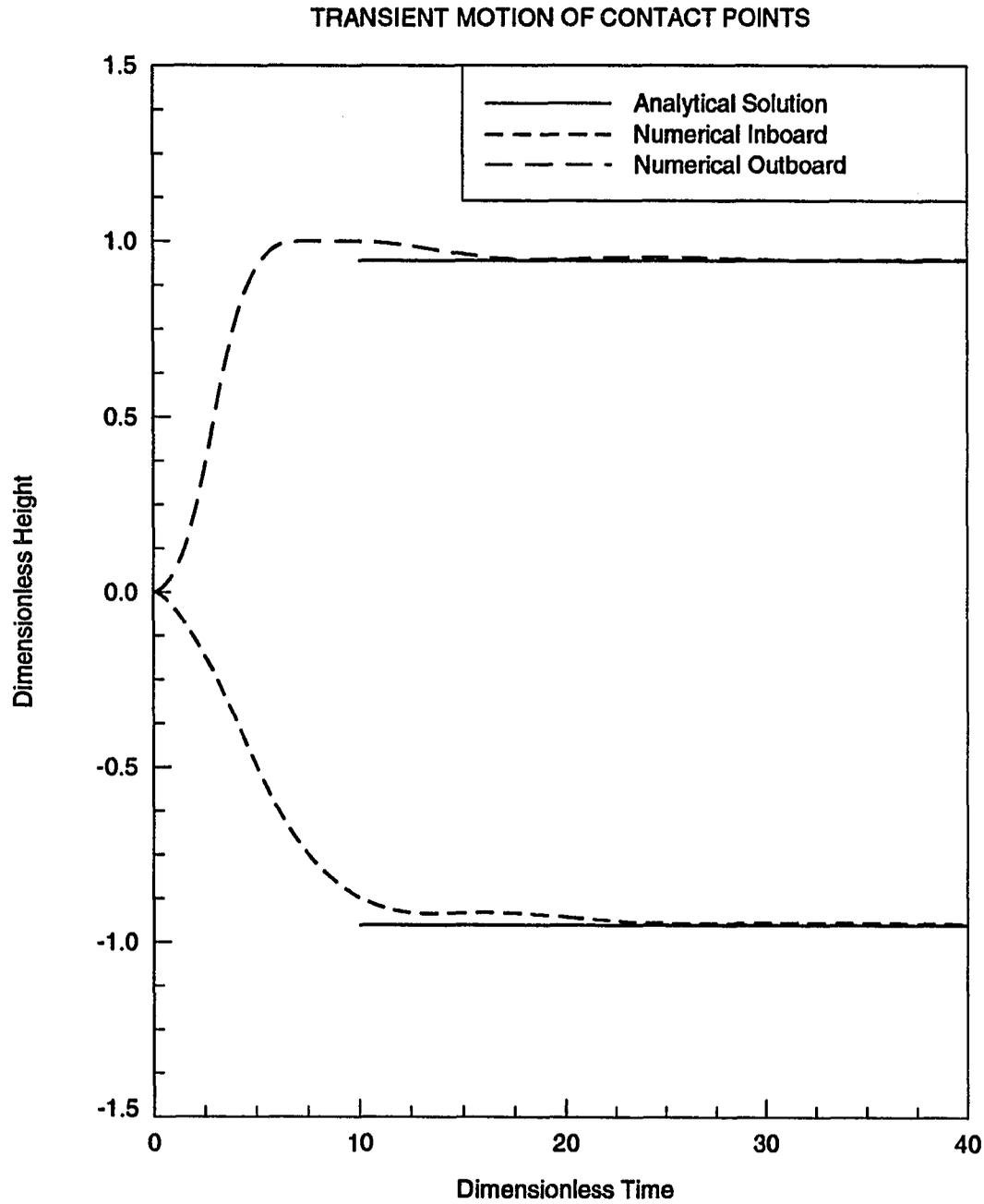


Figure 3.9: Transient motion of the inboard and outboard free-surface contact points, including analytical steady state values, $Re = 300$

STEADY STATE FREE SURFACE PROFILES

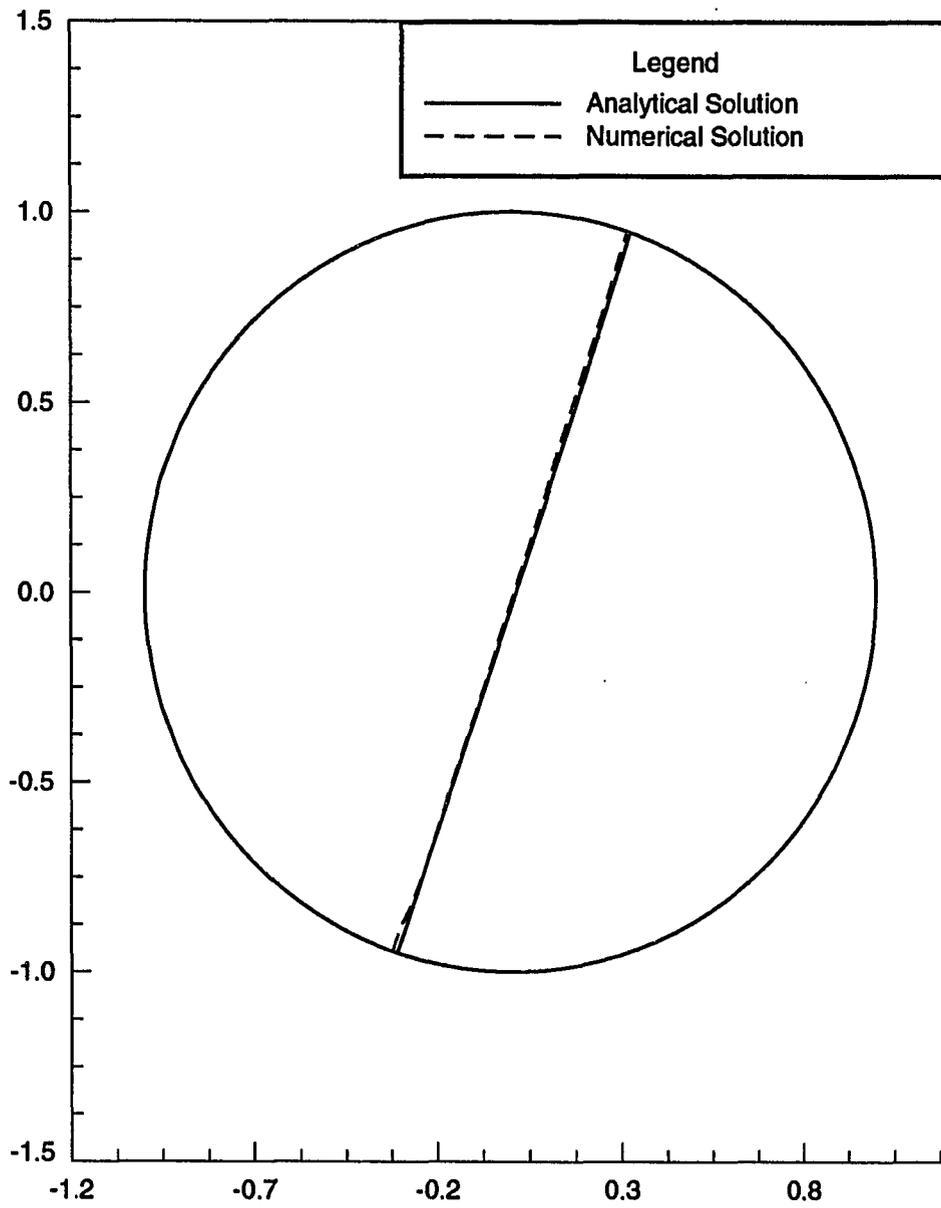


Figure 3.10: Comparison of the analytical and numerical steady state free-surface profile in the plane $x_{32} = 0$ for the $Re = 300$ case

4. VERIFICATION OF TRANSIENT ACCURACY OF THE SURFACE-FITTING APPROACH

It is very important in the simulation of unsteady fluid flows to thoroughly verify the transient accuracy of the computed results. The difficulty in verifying the results of free-surface flow computations is in the lack of good experimental data. Martin and Moyce [52] have recorded some useful data that has served as a benchmark for verifying the accuracy of free-surface algorithms used in other investigations [56] [42].

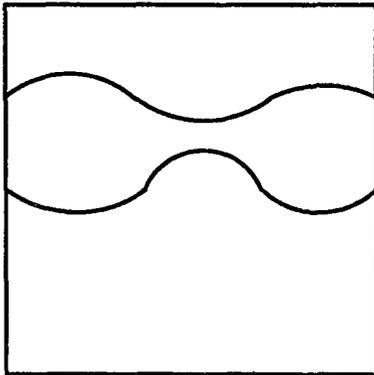
4.1 The three-dimensional broken dam problem

Several investigators have analyzed the motion of falling fluid columns of different cross sections [53]. The experiments conducted by Martin and Moyce as part of these studies pertain to the broken dam problem. Though their data included liquid columns of axisymmetric as well as rectangular cross-sections, the plane symmetric rectangular case has been used for the validation of the present code.

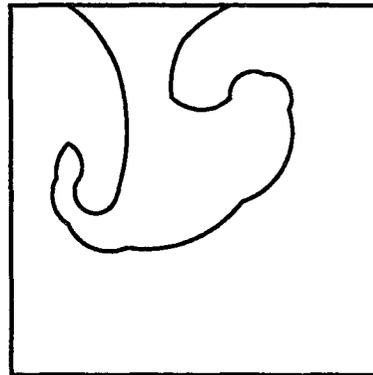
The problem consists of a column of liquid with a rectangular cross-section; one face of the column is suddenly removed and the fluid allowed to flow instantaneously through a channel whose width equals the width of the original column. Experimental measurements include the transient position of the surge front as well as the height of the falling column of liquid as a function of time.

4.2 Free-surface treatment: surface method 'C'

Certain key modifications have been made to the computational procedure outlined in the case of sloshing of liquids in spherical containers. These include: a) a capability to subdivide the free-surface into more than one piece with each piece assigned to a different free-surface definition and b) further refinement to the free surface calculation procedure described earlier. The next few paragraphs shall describe these changes in detail.



Single valued



Not single valued

Figure 4.1: Examples of free-surface shapes that satisfy as well as that do not satisfy the single-valued requirement

The free surface kinematic equation originates from the classical free-surface boundary condition that states that a particle on the free-surface remains on the surface [58]. The mathematical equality requires the free-surface function to be a single valued dependent of the spatial variables at any instant in time. Generally the free-surface is defined as the value of height (z) at various points in the (x,y) plane. For example, $F = z(x,y)$ is a very common way of defining three-dimensional

surfaces. Examples of two-dimensional free-surface shapes that satisfy this single valued requirement as well as ones that do not are shown in the Figure 4.1. In applications with low to moderate surface deformations, it is reasonable to expect a single function spanning the entire free surface to be single valued. The same cannot be true in many instances like the very early duration in the broken dam problem and liquid inside partially filled containers at very low levels of gravity where bubble shaped geometries are fairly common. Another example could be the falling (or rising) of a drop of one fluid through another fluid. It would be very difficult to define free-surface functions that remain single valued and span the entire surface. This problem has been overcome in the present study by splitting the free-surface into parts that can be defined as single valued functions of spatial variables.

The free-surface definitions used for the broken dam problem is shown in Fig 4.2. The top picture shows the initial position of the fluid column and the bottom picture shows the fluid at a later instance. The free surface is divided into two parts: F_1 which is a function of x_1 and x_2 , and F_2 which is a function of x_1 and x_3 . F_1 is the value of x_3 at different locations defined by x_1 and x_2 . Similarly F_2 is the value of x_2 coordinate at different (x_1, x_3) locations. It can be seen that other ways of defining the whole free-surface as one piece could be cumbersome for this problem.

The free-surface kinematic equations that are solved to obtain the solutions F_1 and F_2 are given by:

$$\frac{\partial F_1}{\partial \tau} = \{u_{33} + f_{3k}x_{3k}\} - \{\dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2}\} \frac{\partial F_1}{\partial z_1}$$

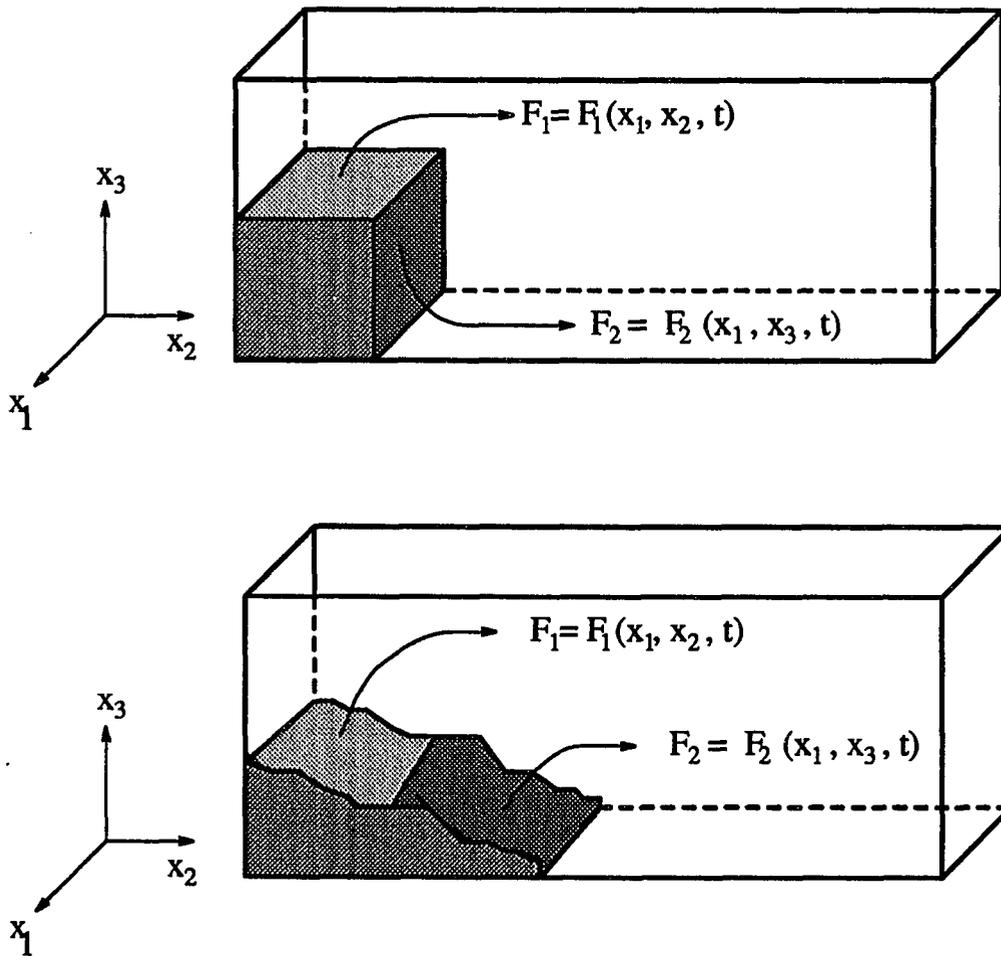


Figure 4.2: Schematic of the three-dimensional broken dam problem

$$-\{\dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2}\} \frac{\partial F_1}{\partial z_3} \quad (4.1)$$

$$\begin{aligned} \frac{\partial F_2}{\partial \tau} &= \{u_{32} + f_{2k}x_{3k}\} \\ &-\{\dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{33} + f_{3k}x_{3k})\eta_{1,3}\} \frac{\partial F_2}{\partial z_1} \\ &-\{\dot{z}_3 + (u_{31} + f_{1k}x_{3k})\eta_{3,1} + (u_{33} + f_{3k}x_{3k})\eta_{3,3}\} \frac{\partial F_2}{\partial z_3} \end{aligned} \quad (4.2)$$

The time derivatives have been treated with a backward difference and the spatial derivatives have been expressed through first-order upwind differences. The upwinding of the spatial terms was found necessary to remove some of the saw tooth like free-surface profiles that result if the same terms were treated with central differences. These oscillatory free surface profiles are highly undesirable as the interior grid lines also tend to follow these shapes resulting in numerical instabilities. As there are two patches of the free-surface, both the equations have to be solved instead of just one equation as was done in earlier sloshing calculations.

The free-surface procedure, method 'B' which was described earlier for the sloshing problem has been further refined for the broken dam problem. The major change follows the realization that the free-surface information could be updated even before convergence has been achieved in the primitive variable iterative updates. In the procedure adopted for the broken dam problem, free-surface equations

$$\begin{aligned} \left[\frac{\partial F_1}{\partial t} \right]^{n+\frac{1}{2}} &= \frac{F_1^{n+1} - F_1^n}{\Delta t} \\ &= \frac{[C_1]^n \left[\frac{\partial F_1}{\partial z_1} \right]^n + [C_1]^{n+1} \left[\frac{\partial F_1}{\partial z_1} \right]^{n+1}}{2} \end{aligned} \quad (4.3)$$

$$+ \frac{[C_2]^n \left[\frac{\partial F_1}{\partial z_2} \right]^n + [C_2]^{n+1} \left[\frac{\partial F_1}{\partial z_2} \right]^{n+1}}{2} + \frac{[S]^n + [S]^{n+1}}{2}$$

where

$$C_1 = \{ \dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{32} + f_{2k}x_{3k})\eta_{1,2} \}$$

$$C_2 = \{ \dot{z}_2 + (u_{31} + f_{1k}x_{3k})\eta_{2,1} + (u_{32} + f_{2k}x_{3k})\eta_{2,2} \}$$

$$S = \{ u_{33} + f_{3k}x_{3k} \}$$

and

$$\begin{aligned} \left[\frac{\partial F_2}{\partial t} \right]^{n+\frac{1}{2}} &= \frac{F_2^{n+1} - F_2^n}{\Delta t} \\ &= \frac{[C_1]^n \left[\frac{\partial F_2}{\partial z_1} \right]^n + [C_1]^{n+1} \left[\frac{\partial F_2}{\partial z_1} \right]^{n+1}}{2} \\ &+ \frac{[C_3]^n \left[\frac{\partial F_2}{\partial z_3} \right]^n + [C_3]^{n+1} \left[\frac{\partial F_2}{\partial z_3} \right]^{n+1}}{2} + \frac{[S]^n + [S]^{n+1}}{2} \end{aligned} \quad (4.4)$$

where

$$C_1 = \{ \dot{z}_1 + (u_{31} + f_{1k}x_{3k})\eta_{1,1} + (u_{33} + f_{3k}x_{3k})\eta_{1,3} \}$$

$$C_3 = \{ \dot{z}_3 + (u_{31} + f_{1k}x_{3k})\eta_{3,1} + (u_{33} + f_{3k}x_{3k})\eta_{3,3} \}$$

$$S = \{ u_{32} + f_{2k}x_{3k} \}$$

are solved after every iterative update of the primitive variables in the CSIP routine. The equations 4.3 and 4.4 approximate the time derivative of the free-surface function at the mid-point between the previous time level and the current time level. This was done by averaging the spatial derivatives evaluated at both the previous and the current time level. In the case of linear systems, such an approximation is second order accurate in time very much similar to the Crank-Nicolson scheme for the heat equation or the trapezoidal differencing for the linear wave equation [50].

In estimating the coefficients for spatial derivatives at the present time ($n + 1$), the most recent information was used in the surface method 'C'. This is in contrast to the method 'B' where these terms were evaluated at the latest instance when primitive variables converged. Subiterations and updates following this convergence make more recent information available. It is this most recent information that is used in method 'C' but not in method 'B'.

Every free surface update necessitates the update of grid speed, metric terms and information on normal and tangent vectors at the surface, for all the points on the free surface. This free-surface update, however, does not require update of grid speed and metric terms in the rest of the grid as these points have not moved. The ideal approach would be to solve the free surface equations every time the velocities have been updated, and continue towards convergence in the primitive variables at the current time level. This process encountered some difficulties owing to the structured grid used in the present study. The grid lines representing the free-surface moved closer and closer to the neighboring grid lines in the interior of the liquid, with every free-surface update. At times, this resulted in grid lines (of different z_i values) crossing each other and led to numerical instabilities and eventually caused the calculations to blow up. This was avoided by generating a grid every few iterations of free-surface updates.

It should be noted that generating new grids should be avoided as much as possible as they amount to lot of additional computational work. The additional work has two components, the first of which follows from the actual grid generation process, and the next from having to update grid speed and metric terms as a result of the newly generated grid. Hence a compromise of generating a grid for every ten

iterations of velocity updates (while solving for new free surfaces following every one of these updates) was implemented in the broken dam calculations.

To summarize, the final procedure, employing the surface method 'C', used for the three-dimensional broken dam calculations consisted of the following steps:

1. Solve the kinematic conditions given by Equations (4.1) and (4.2) for the free surface position
2. Generate a grid, compute grid speed, metric terms etc.
3. Compute elements of the matrix A, and vector 'b' in $A\vec{q} = \vec{b}$
4. Calculate increments for the primitive variables by CSIP
5. Update the primitive variables
6. Check for convergence
7. IF not converged
 - (a) Solve for revised free-surface information by using the Eqs. (4.3) and (4.4).
 - (b) Update free-surface normal and tangent vectors;
 - (c) Update grid speed and metric terms for points on the free-surface
 - (d) IF (new grid is necessary / predetermined number of iterations have passed since last grid generation) then generate a grid, compute grid speed, metric terms etc. for interior points as well
 - (e) Compute elements of the matrix A, and vector 'b' in $A\vec{q} = \vec{b}$
 - (f) Calculate increments for the primitive variables by CSIP

(g) Update the primitive variables

(h) GO TO step 6

ELSE (i.e. if convergence has been achieved)

(a) increment time-level; go to step 1.

ENDIF

The surface method 'C' described above is superior to the methods 'A' and 'B' explained earlier. This has also been found to give the best results among the three methods. For this reason, it is recommended as the one to be used for future free-surface calculations using the surface-fitting approach.

4.3 Discretization of the momentum equations

It was believed that to get accurate transient results, the momentum equations should be discretized to higher-order temporal accuracy in contrast to the first-order expressions listed in Chapter two. Hence the following changes were made to the approximations for terms in the momentum equations.

In the sloshing calculations described so far, the time derivative term was differenced with a backward difference involving the present time level and the previous one. The spatial derivatives were differenced using second order central differences. Similarly, the time derivative in grid speed terms were treated with a backward difference in time, and second order central differencing was used for spatial difference terms in the metric derivatives.

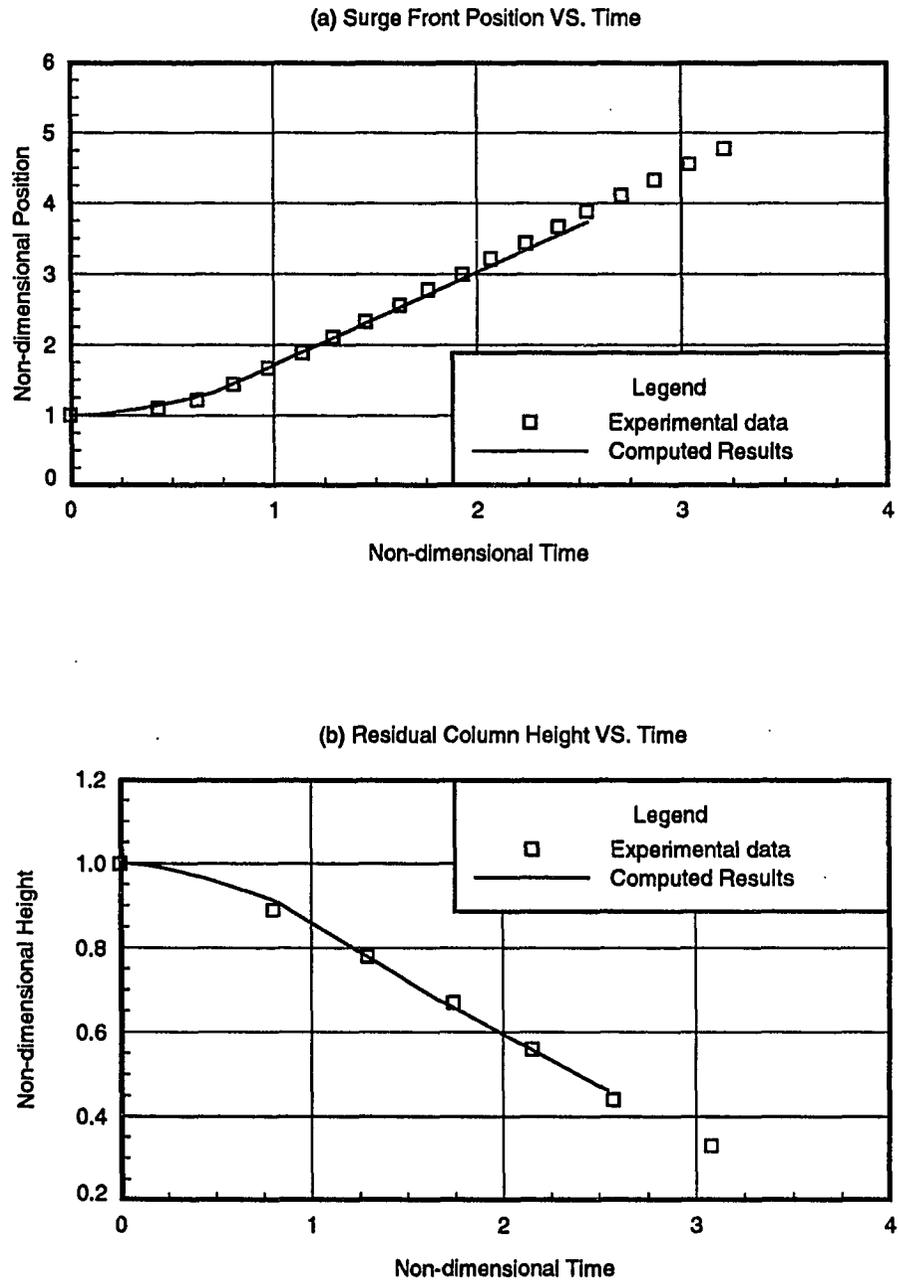


Figure 4.3: Comparison of numerical and experimental transient data for the three-dimensional broken dam calculations

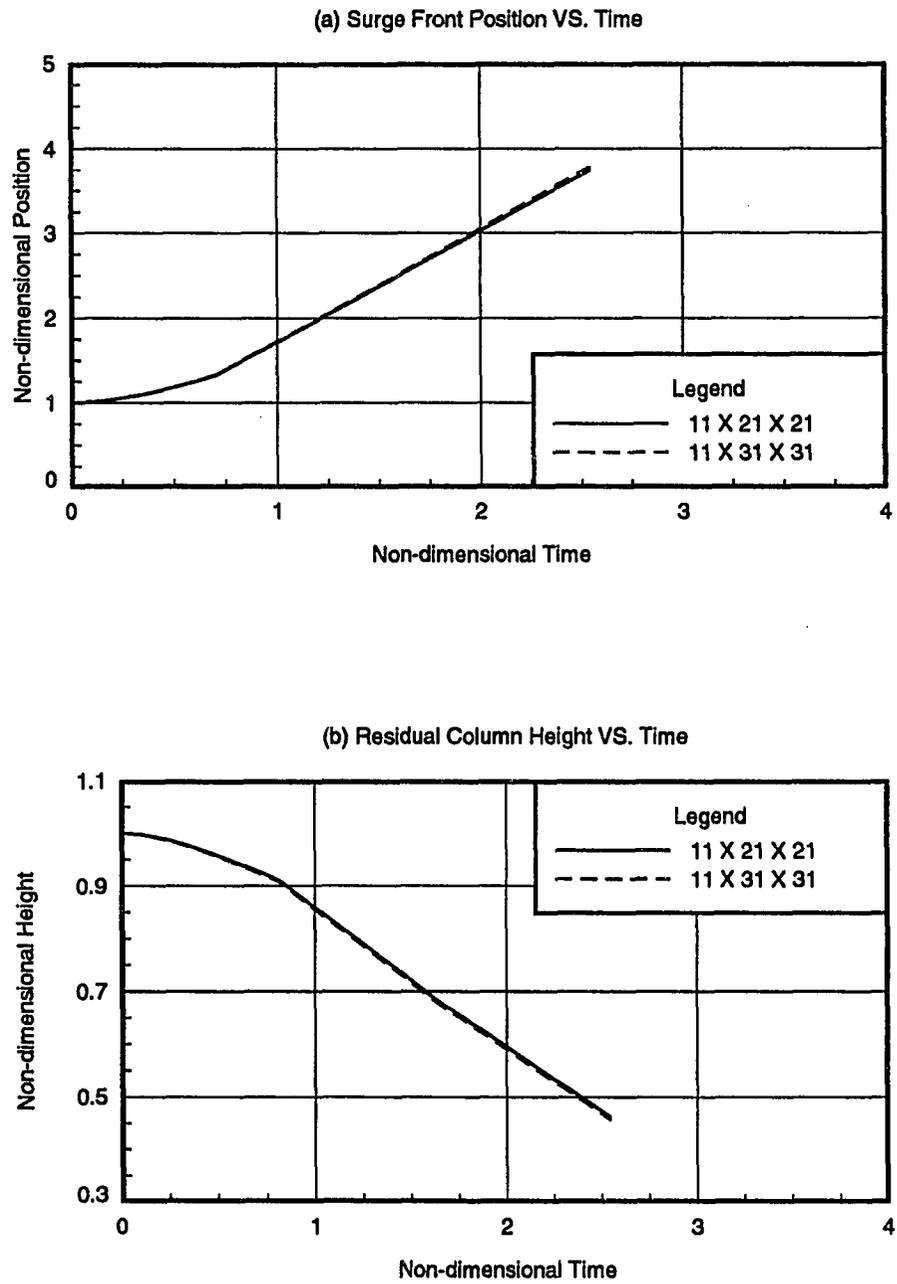


Figure 4.4: Verification of grid independence of the broken dam free-surface calculations

As can be seen, the time discretization uses only two time levels and hence is of first order accurate in a conventional sense. To improve this aspect of the accuracy the differencing was changed to make it closer to being second-order accurate in time, by following the Crank-Nicolson type differencing for all the terms other than those that involve a partial derivative with respect to time. This is similar to the averaging between two time levels, much along the lines of the time-centered implicit differencing explained with respect to the free-surface kinematic equation. Understanding this can be made much simpler by looking at a generic form of the momentum equations:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - g_i \quad (4.5)$$

After linearizing, this equation could be rewritten as:

$$\frac{u^{n+1} - u^n}{\Delta t} = C_1 \frac{\partial u}{\partial x_j} + C_2 \frac{\partial^2 u}{\partial x_j \partial x_j} + S = RHS \quad (4.6)$$

The whole right hand side (RHS) was evaluated as an average between the two time levels, n and $n + 1$. That is,

$$RHS = \frac{(RHS)^{n+1} + (RHS)^n}{2}$$

Caution is generally advised in using such averaging procedures in the case of moving grids, such as those used in the present problem. As the governing equations are used in the chain rule, non conservation law form [63], such an averaging is possible even though the points involved in the averaging process are at different physical locations at the two time instances. In the present formulation, the grid

motion is properly accounted for through the grid speed terms. Such a procedure may not always be allowed in some adaptive and moving grid schemes.

4.4 Results for the three-dimensional broken dam calculations

The initial geometry of the fluid column was a cube whose edges were 2.25 inches long. The fluid was water and the nondimensional parameters used in the plots are the same as used by Martin and Moyce [52]. The comparison of the transient motion of the surge front and the falling column height with the experimental data is shown in the Fig 4.3. These results seem to be in good agreement with the experimental data. The calculations could not be continued owing to restrictions on computing resources. These calculations were performed on Cray Y-MP supercomputers and took about 180 minutes for each calculation. The procedure has been checked for grid size dependence and the results are shown in the Fig 4.4. The direction in which 11 points has been used is the one that is normal to the main flow directions, i.e. the x_1 direction of Fig 4.2. The number of points in this dimension was not increased in the finer grid calculations as there was not any major flow pattern to be resolved in that direction. This calculation utilized more than 10,000 points and took about 310 minutes on the Cray Y-MP.

A comparison of the transient motion of the surge front computed by the new procedure 'C' and the method used earlier for the calculations (method 'B') is also shown in Fig 4.5. It is very clear that the present procedure is superior. It should however be noted that the results for the sloshing in spherical containers discussed earlier are believed to be time accurate for the following reasons:

1. Good agreement with experimental data has been observed [51]

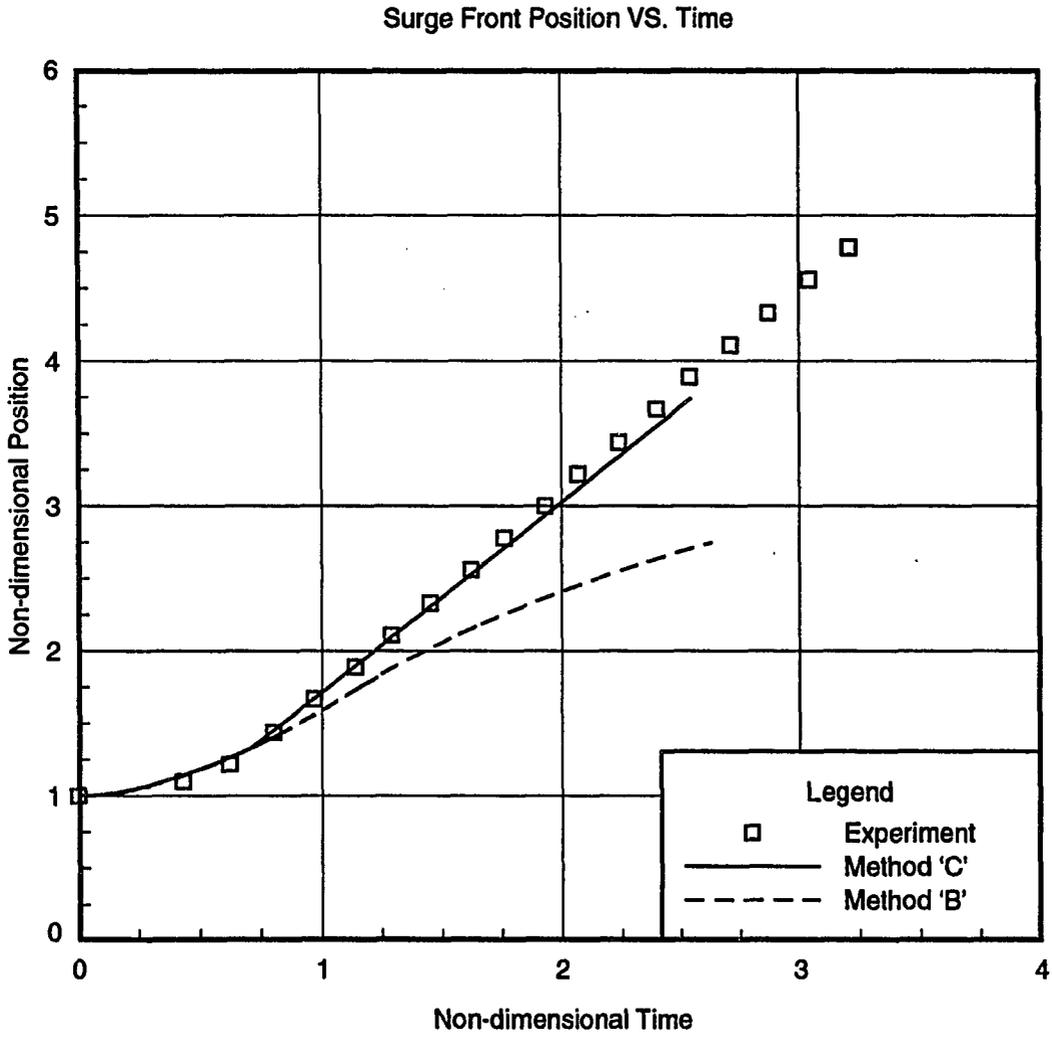
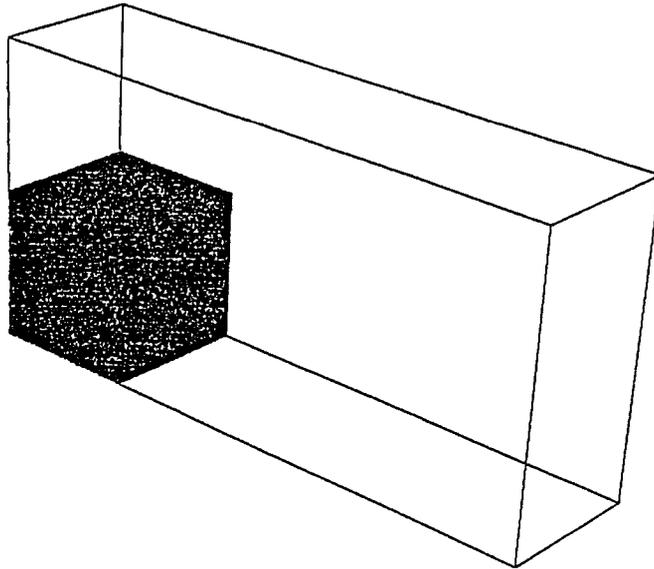


Figure 4.5: Comparison of experimental data on the transient surge front motion with numerical results from two different procedures

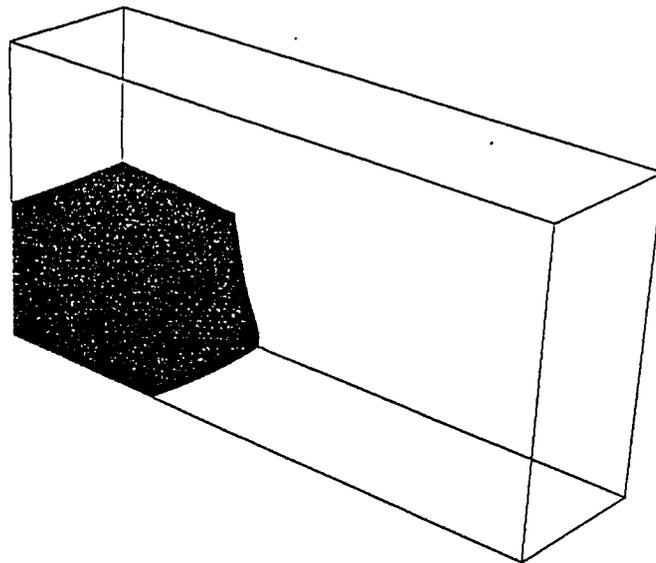
2. The better temporal resolution of the free-surface motion is believed to be more important in the case of low viscosity (and hence high Reynolds number) fluid flows

Another reason for the method 'B' not performing so well in the case of the broken dam problem could be due to the larger deformation in this problem. This in turn results in significantly more stretching of the grid cells. The fluid was contained in the sloshing problems and the movement of the free-surface before it reached steady state was not significant, though the surface underwent considerable warping. This claim has the added support that the method 'B' still produced good results for the early part of the broken dam problem. It can be seen from Figure 4.5 that the results from the earlier approach 'B' agrees well with the experimental data until the surge front travels to a position where the length of the bottom fluid plane is about 1.5 times its initial length.

The position of the flowing liquid at various select instances is shown in Figure 4.6. The velocity vectors in $i = \text{constant}$ and $j = \text{constant}$ planes are shown in Figures 4.7 and 4.8. The shapes of the flowing liquid are also qualitatively similar to the photographs published along with the experimental results of Martin and Moyce [52]. The velocity vector plots are also qualitatively consistent with the general flow pattern.

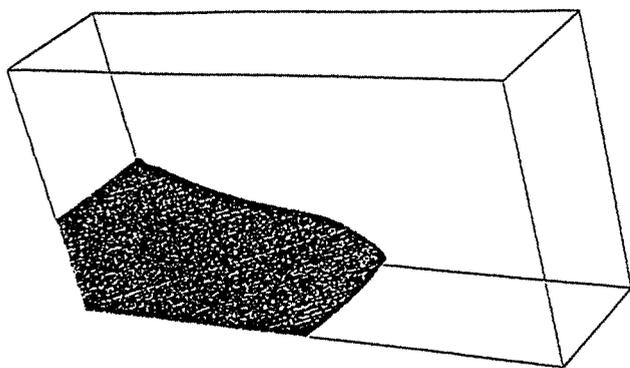


Position of liquid at time = 0.0

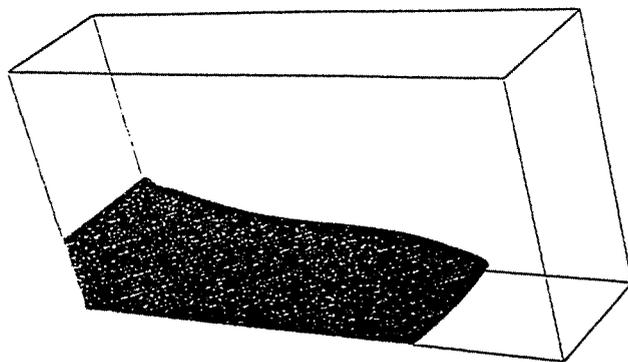


Position of liquid at time = 0.8

Figure 4.6: Position of flowing liquid at select instances

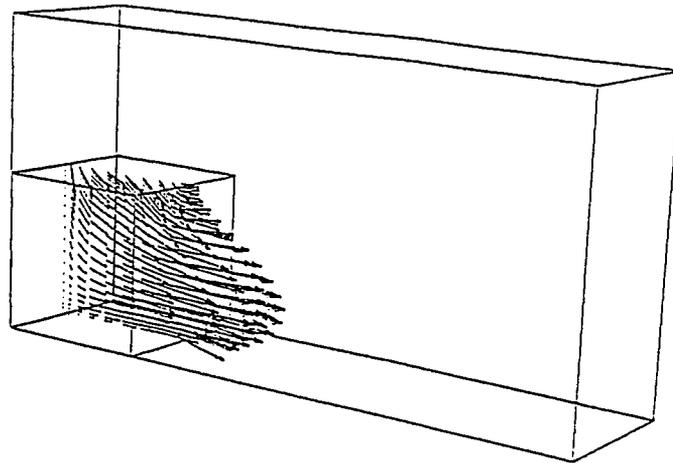


Position of liquid at time = 1.6

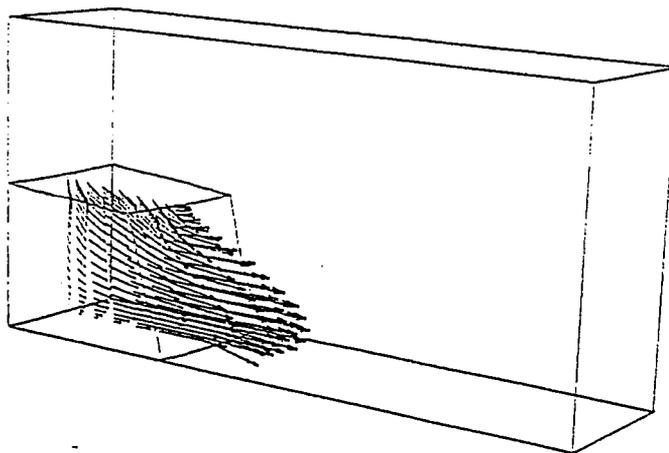


Position of liquid at time = 2.4

Figure 4.6 (Continued)
Position of flowing liquid at select instances

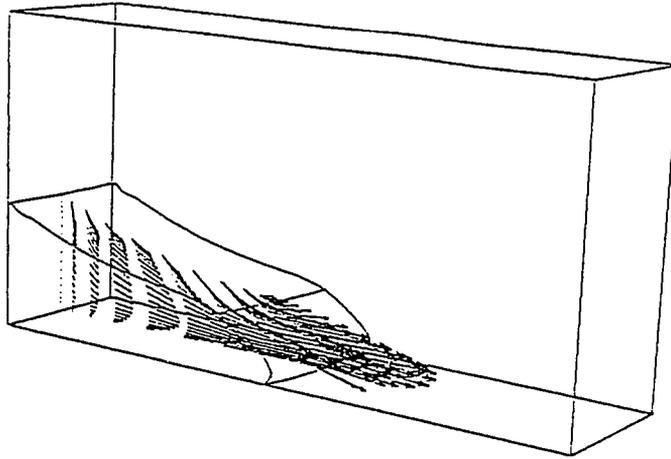


Mid-Plane Velocity Vectors at time = 0.1

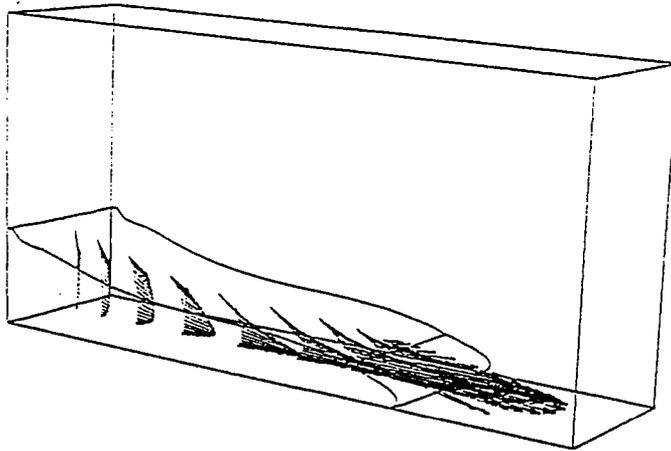


Mid-Plane Velocity Vectors at time = 0.8

Figure 4.7: Velocity vector plots in the $x_1 = 0$ plane at select instances



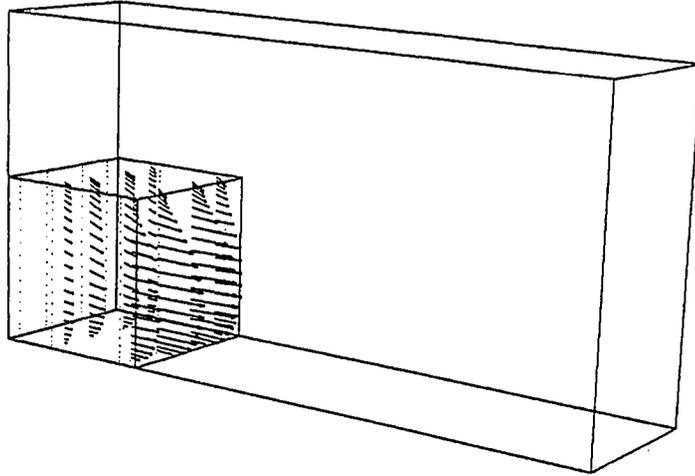
Mid-Plane Velocity Vectors at time = 1.6



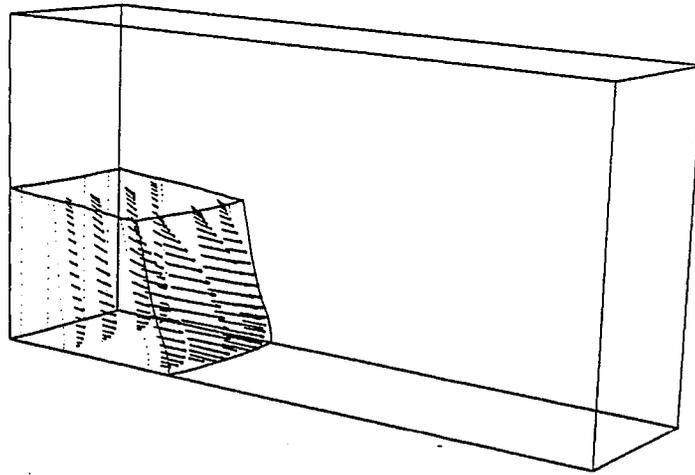
Mid-Plane Velocity Vectors at time = 2.4

Figure 4.7 (Continued)

Velocity vector plots in the $x_1 = 0$ plane at select instances

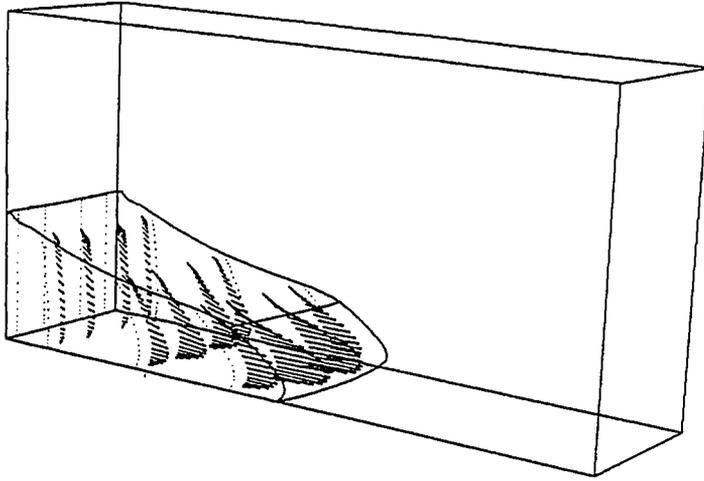


Cross-Plane Velocity Vectors at time = 0.1

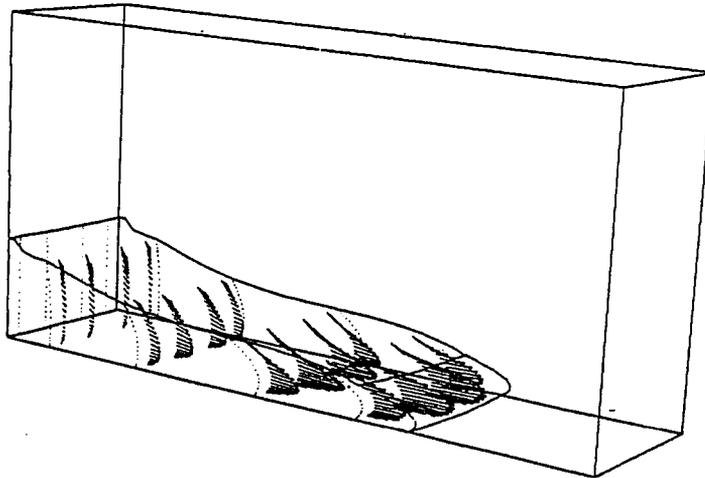


Cross-Plane Velocity Vectors at time = 0.8

Figure 4.8: Velocity vector plots on $z_2 = \text{constant}$ planes at select instances



Cross-Plane Velocity Vectors at time = 1.6



Cross-Plane Velocity Vectors at time = 2.4

Figure 4.8 (Continued)

Velocity vector plots on $z_2 = \text{constant}$ planes at select instances

5. EFFECTS OF LOW GRAVITY

Liquid sloshing flows under the influence of gravity levels much lower than on the earth have received significant attention in recent years. There are many instances where such flow phenomena play a significant role in space systems, most typical of them being fuel storage and handling systems such as those in satellites and the space station.

Leslie [67] performed a series of measurements on rotating equilibrium, free surface shapes in the microgravity environment of a free falling aircraft. Hung et al. have performed a series of low and microgravity calculations of both equilibrium surface profiles and transient free surface flows of fluids in a partially filled cylinder [68] [69] [70].

The formulation used in the present study for the liquid sloshing flows is capable of handling different levels of gravity and also significant surface tension effects. The purpose of this part of the present research effort is to test this capability. Axisymmetric spin-up of a cylinder filled to 0.8 times its initial height with a liquid has been studied at two different levels of gravity (1 g and 10^{-3} g). Prior to this calculation, all the liquid sloshing computations have dealt only with spherical containers. A simple axisymmetric spin-up of a cylinder partially filled with ethanol was performed to verify the computer code for the new tank shape. The numerical and analytical steady

state free-surface profiles for this computation is shown in Figure 5.1. The transient motion of the free-surface position at the tank axis is also shown in Figure 5.2.

During the low gravity calculations, it was also planned to verify if all the nondimensional parameters governing such flows have been properly identified. Two cases of spin-up of partially filled cylinders were chosen so that the physical parameters like the gravity, rotational speed, viscosity and the coefficient of surface tension would be different but the three nondimensional parameters (Re , Fr , and We) had the same values. The Reynolds number, Re , the Weber number, We have been defined previously. The Froude number, Fr is defined as follows:

$$Fr = \frac{V_{ref}}{\sqrt{gh}}$$

where ' h ' is the initial free-surface depth and ' g ' the gravitational acceleration. It was found that the two cases produced identical transient results. The dimensionless parameters governing the sloshing of liquids in containers are listed in Table 5.1. The properties and the values of dimensionless parameters for these two cases are given in Table 5.2. The identical results from the computations do in fact point out that the dimensionless parameters governing the problem have been properly identified. A more formal method of arriving at these parameters is through a dimensional analysis. Such an analysis for this problem can be found in Appendix D. The results from this analysis also agrees well with the earlier conclusion from the computational analysis.

The transient motion of the free surface point on the axis of the cylinder for both cases are given in Figure 5.3.

Steady State Free Surface Profile

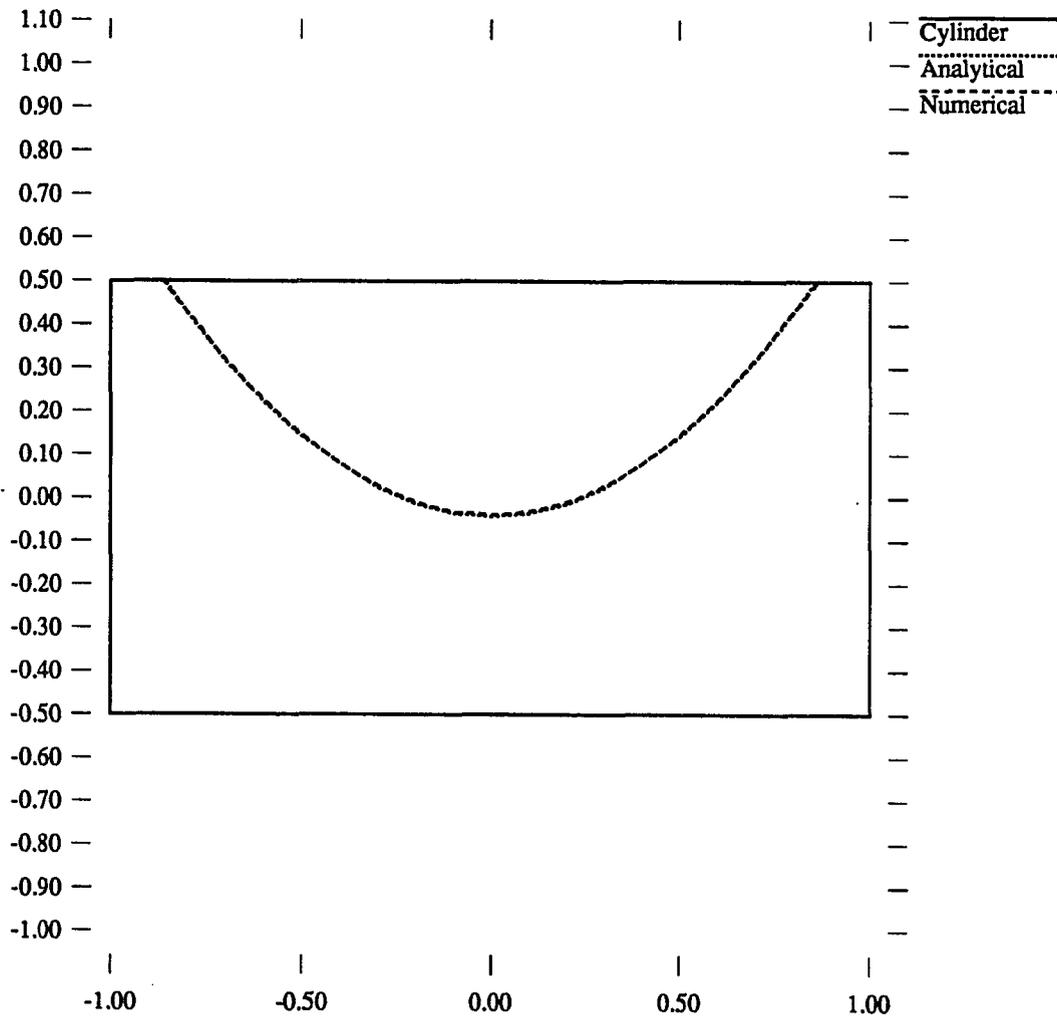


Figure 5.1: Free-surface profiles at steady state

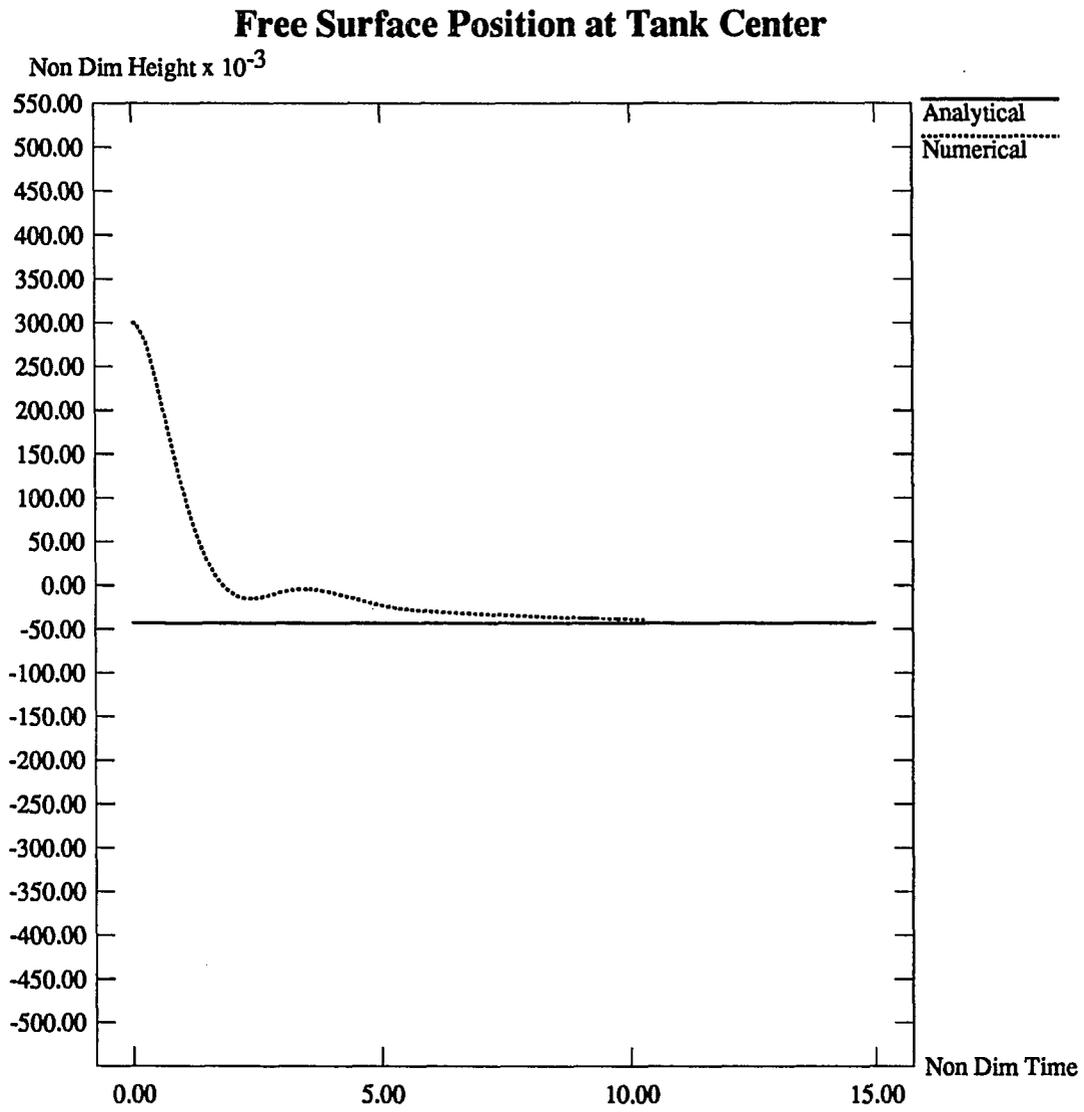


Figure 5.2: Transient motion of the free-surface position at tank center

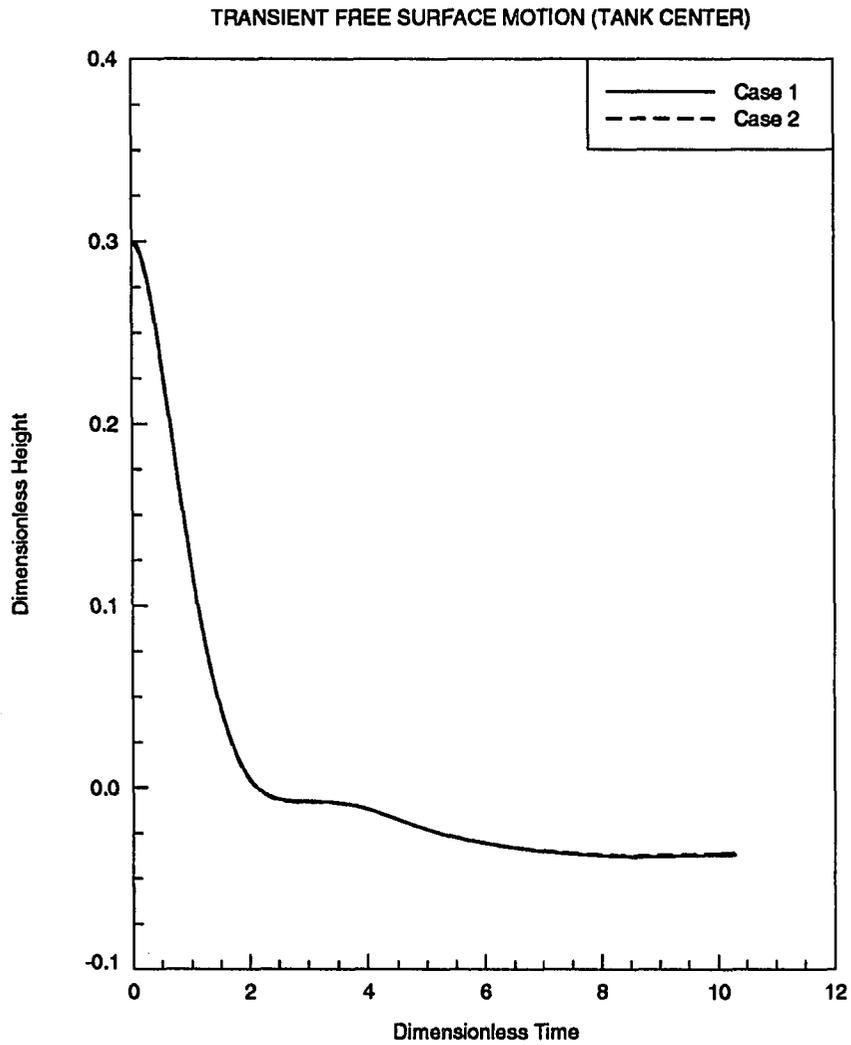


Figure 5.3: Comparison of transient motion of the free-surface position at tank center for the two cases

Table 5.1: Dimensionless parameters governing the axisymmetric sloshing motion inside cylinders

Parameter	Definition
Re, Reynolds number	$\frac{V_{ref} L_{ref}}{\nu}$
Fr, Froude number	$\frac{V_{ref}}{\sqrt{gh}}$
We, Weber number	$\frac{\rho V_{ref}^2 L_{ref}}{\Gamma}$
Height ratio	$\frac{h}{L_{ref}}$

5.1 Discussion and Results

The two cases were chosen so that the final steady state free-surface profiles are identical in both cases with the fluid touching the top wall of the cylinder at steady state. The computed steady state free surface profiles agree very well with the analytical solution obtained by ignoring surface tension effects. It can be seen from Table 5.1 that the Weber number, We is a measure of relative effect of surface tension and inertial forces. A larger value denotes a diminished surface tension effect. The Weber number had values of the order of 10^3 for the calculations of flows under the effect of terrestrial gravity (1 g). The value of We for the two cases described in Table 5.2 was significantly lower and had a value of around 60. The final steady state profiles and the transient motion of the free-surface at the tank center were shown previously in the Figures 5.1 and 5.2 for these cylindrical calculations. The steady state free-surface profiles agree well with analytical results. The analytical free-surface profiles were calculated by ignoring surface tension effects as outlined in Appendix E. The free-

Comparison of the various parameters for the two cases

Parameter	Case, 1	Case, 2	Ratio Case 1 / Case 2
Rotational Speed	167.5 rpm	16.75 rpm	10
Gravity	9.81 units	0.0981 units	100
Viscosity	1.18d-3	1.18d-4	10
Surface Tension	6.33d-2	6.33d-4	100

Dimensionless Parameters

Reynolds Number	32.84	32.84	1.0
Froude Number	1.21	1.21	1.0
Weber Number	638.13	638.13	1.0
Ratio of Initial depth to radius	0.8	0.8	1.0

Table 5.2: Comparison of the various parameters for the two cases

surface position at the tank center also reaches its corresponding analytical steady state position. This case has also been computed with an impulsively started initial condition. The two cases explained earlier used the 'initially capped' type of starting conditions. In such a start-up situation, the liquid is assumed to have achieved solid body rotation at the corresponding rotational speed with the free-surface remaining in its horizontal position corresponding to zero rotational speed. The imaginary cap holding the free surface is released at time 'zero' and the surface is allowed to move towards its steady state position. In the impulsively started class of spin-up cases, the fluid is assumed to be at rest and at the beginning of the calculations particles in contact with the wall are subject to a sudden velocity corresponding to the linear speed due to the rotation of the cylinder. As the problem has been formulated in terms of velocities relative to the rotating cylinder, zero velocity conditions are imposed at the wall and velocities equal to the negative of the imposed angular velocities are specified at other points in the fluid. The free surface profile at steady state from this impulsively started condition also agreed well with the analytical solution as shown in Figure 5.4. The transient history of the free surface motion at the tank center for both types of initial conditions is also shown in Figure 5.5. It can be seen that the overshoot from the initially capped start-up is more than that from the impulsively started case. This observation is similar to the results produced earlier by Chen [17] for sloshing in spherical containers. Computations of cases with more prominent surface tension forces are discussed in the next section.

Another new feature of these cylindrical spin-up calculations is the motion of free-surface past interior corners (the junction of the curved cylindrical surface and the top end disc of the cylinder). This feature has not been demonstrated in our

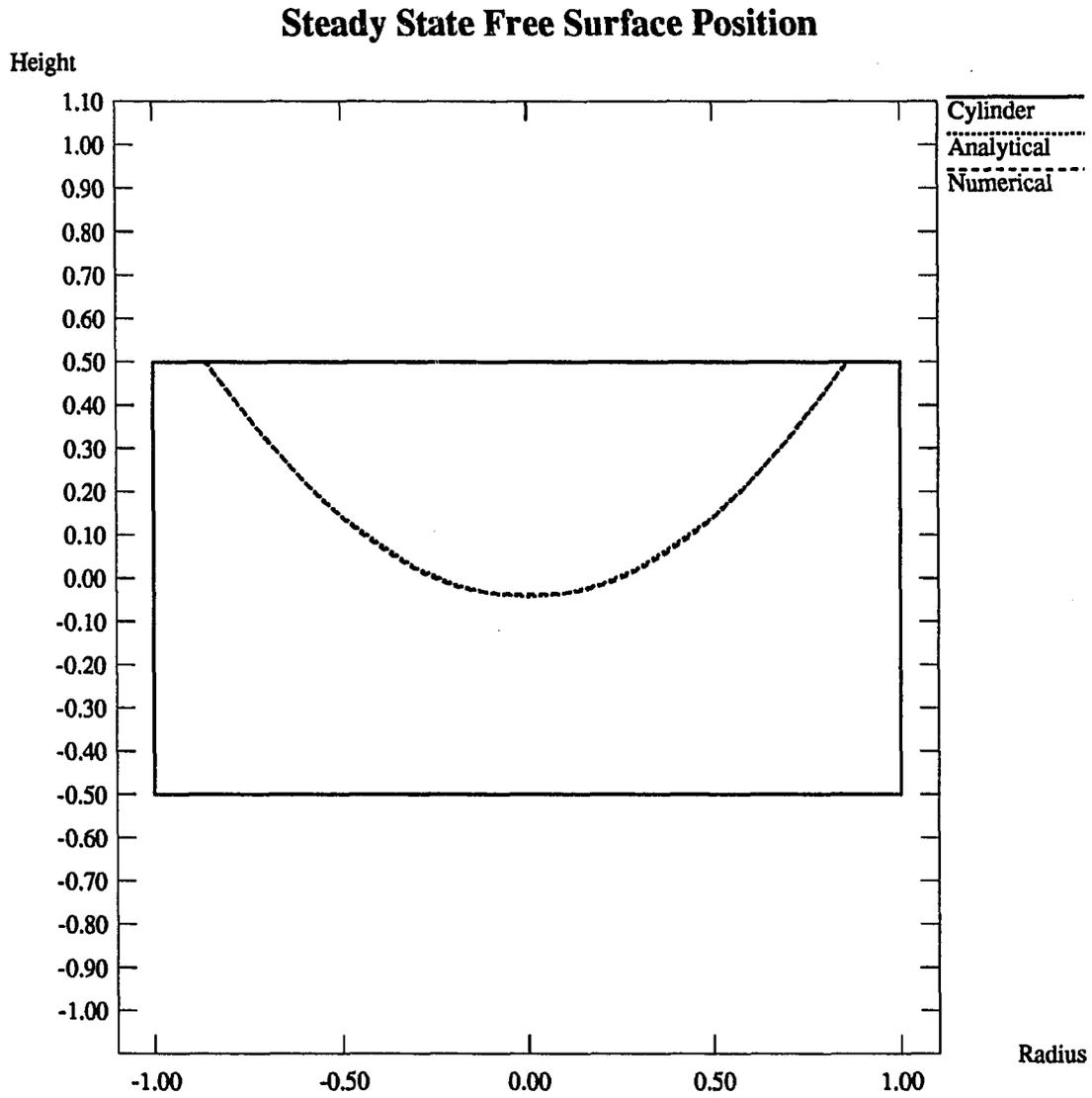


Figure 5.4: Free-surface profiles at steady state, Impulsive start-up

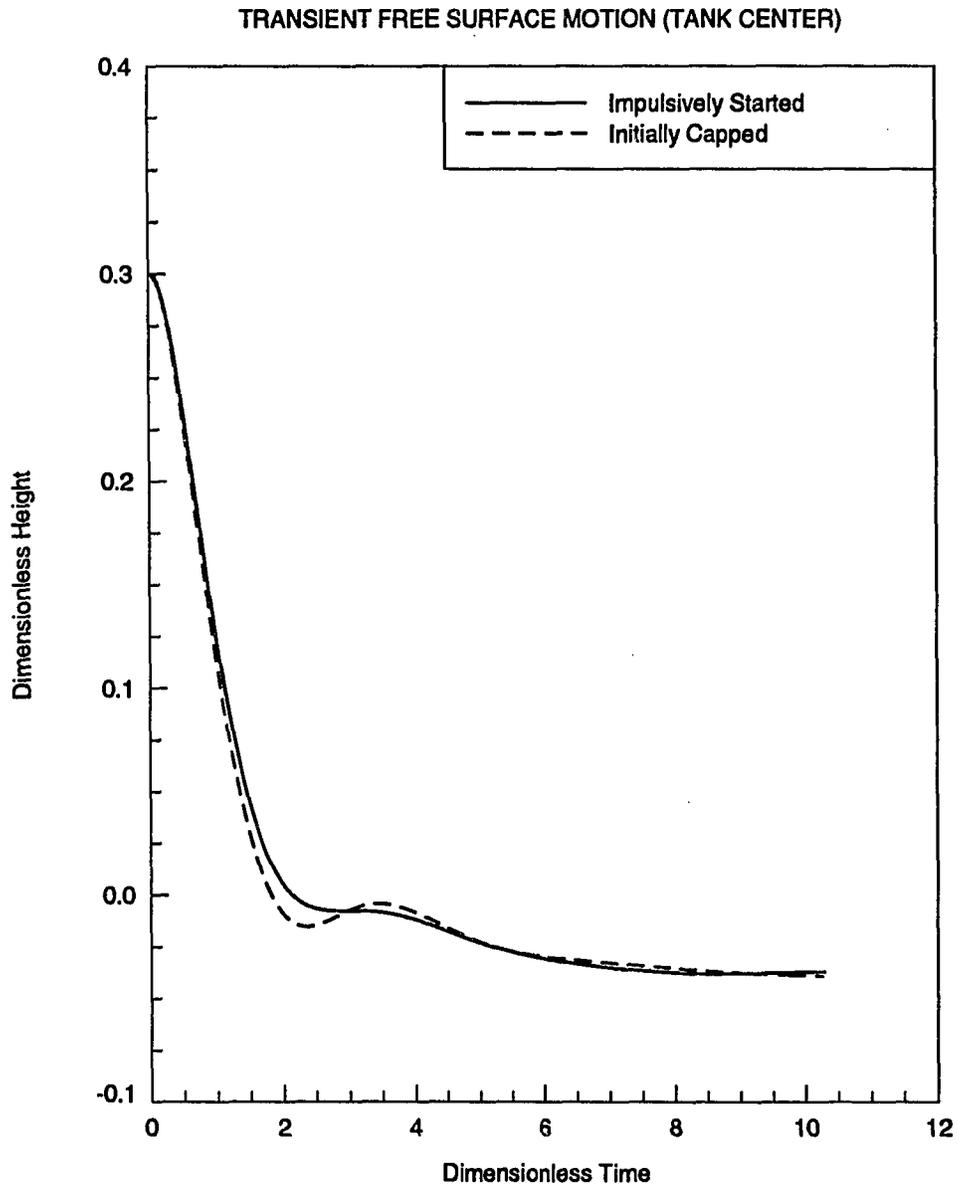


Figure 5.5: Comparison of transient motion of the free-surface position at tank center for the initially capped and the impulsively started calculations

surface fitting approach before. The computer code was modified to recognize the fact that the free-surface may not span the entire cylinder radius. Free-surface boundary conditions were used only up to the point where the surface intersects the top wall and the usual wall boundary conditions were applied on points radially beyond this point where the free-surface touches the top end of the cylinder.

5.2 Calculations of zero gravity flows

Chandrasekhar estimated the equilibrium shapes of rotating drops at zero gravity based on Laplace's equation for the pressure drop across the interface [55]. More recently, experiments of cylinders partially filled with a liquid and rotating in the absence of gravity have been performed and checked against this analytical model [67]. The experimental data have been found to be in good agreement with the analytical results estimated using a generalization of Chandrasekhar's formula. This extension enables including the contact of the interface with the container wall at a specified angle.

The experiments were performed at different values of R , which represents the relative effects of centrifugal force and surface tension. Larger values of F indicate increasing centrifugal effects. The experiments were conducted using a KC-135 aircraft, which flew in a parabolic trajectory providing low gravity levels lasting for 20-30 seconds. A plexiglass cylinder 20 cm in diameter and whose height can set at 2, 4 or 6.3 cm was used. Ethanol was chosen as the fluid owing to its high surface tension, low sensitivity to contamination and surface wetting properties. The contact angle for ethanol was near zero. Overhead and side mounted video cameras recorded the shapes of the fluid.

One of the experimental runs was used to compare with the calculations using the present surface fitting procedure. This run corresponded to a value of $F = 1.1$ with the centrifugal and surface tension forces being nearly equal. The initial free surface profile was chosen as a vertical cylindrical plane parallel to the outer rotating cylinder with the liquid uniformly distributed in the circumferential direction. As time progressed, the free surface moved towards its final position. As this motion was axisymmetric, surface profiles at different instances in one circumferential plane are shown in Figure 5.6. The final numerical results as well as experimental data on the steady state free-surface profiles are shown in Figure 5.7.

These results are very encouraging. More low and microgravity calculations of both axisymmetric as well as three-dimensional situations are necessary to gain further confidence. Only after successful calculations of such flows the algorithm can be declared to possess the capability to compute low gravity flows.

INTERMEDIATE FREE SURFACE PROFILES

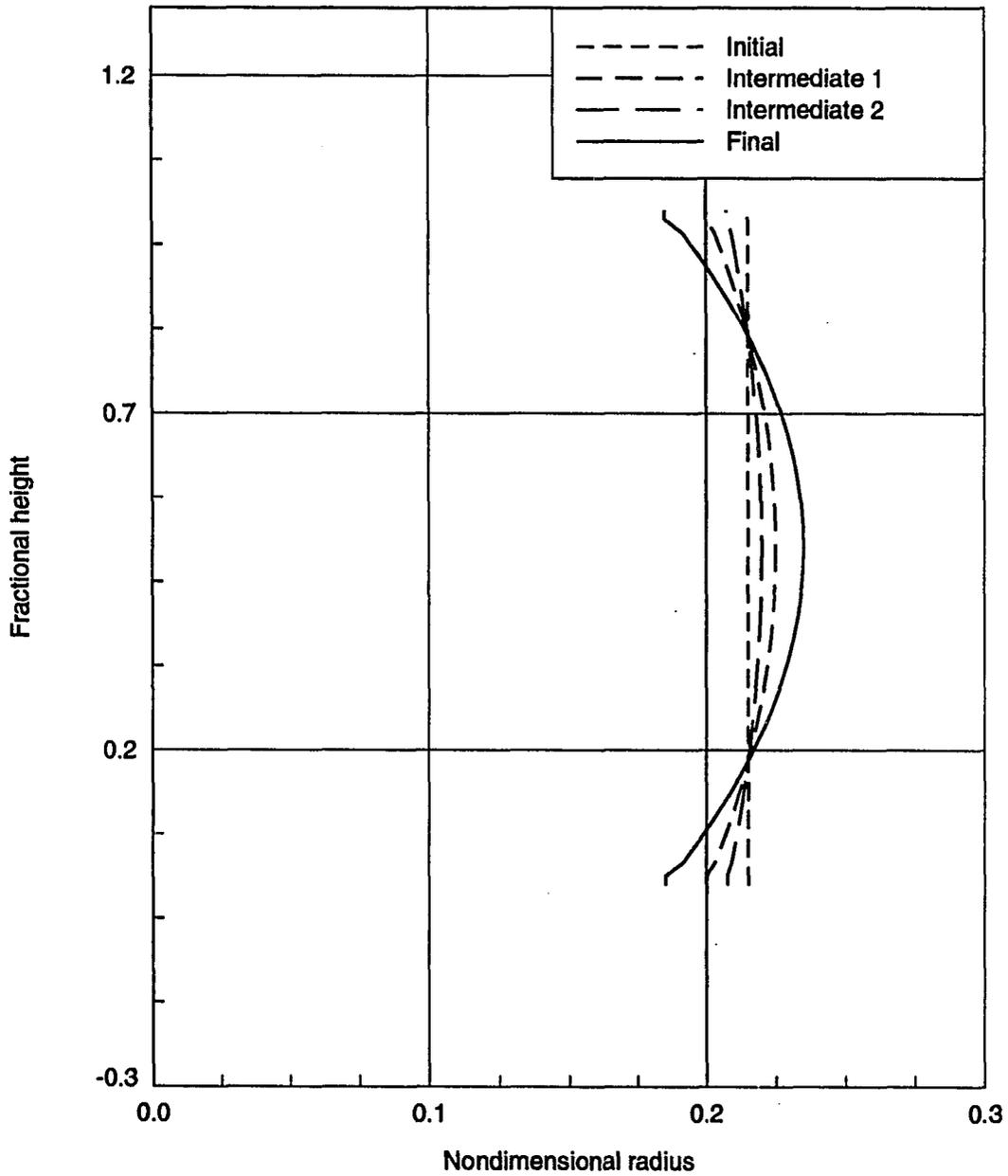


Figure 5.6: Intermediate free-surface profiles during the transient motion from the initial condition to the steady state

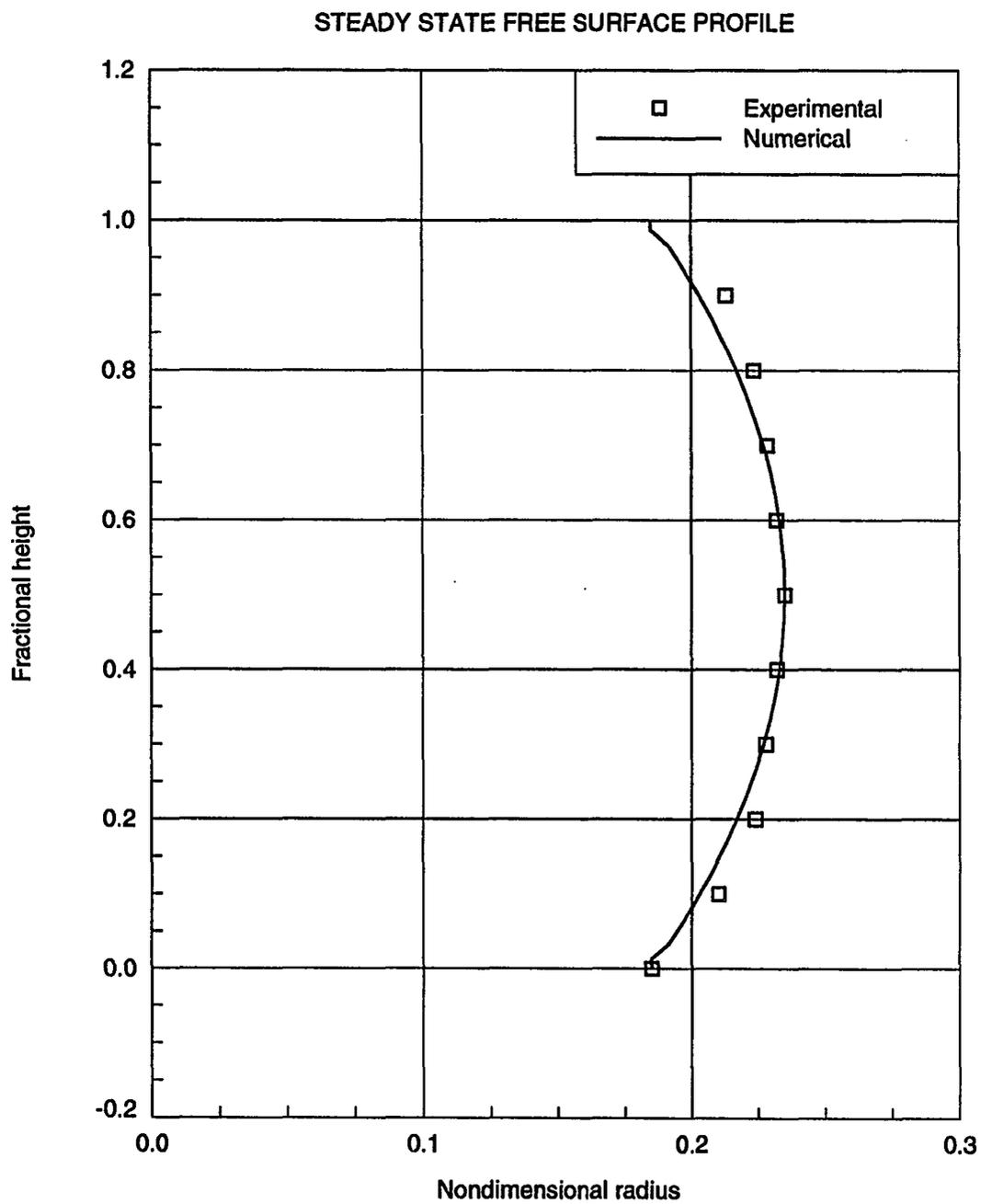


Figure 5.7: Free-surface profiles at steady state

6. FLUID-STRUCTURE INTERACTION

Sloshing fluid stores have been identified as the likely source of stability problems resulting in nutation and tumbling of spin-stabilized satellites. This area has been under investigation at Iowa State University through both experiments and numerical modelling.

Having developed the capability to predict the structural dynamics and the fluid sloshing, it was recognized that coupling the two calculations could provide a better understanding of the phenomena giving rise to the unstable motion. The following sections discuss the early efforts at integrating the two calculations and the initial results.

6.1 Introduction

As part of the research on liquid sloshing in spin-stabilized satellites, two codes have been developed: the first one(STRUCTURE) [37] [38] calculates the flexible system dynamics and the second computes the sloshing motion of the fluid in the tank. The liquid sloshing code used the surface fitting approach and employed surface method 'B' for calculating the free-surface motion. This code has been designated by the name 'SLOSH3D' in chapter one and shall be referred to by this name throughout this chapter. Initially, these two codes have been run as independent entities(non-

interaction mode). Subsequently, the two codes have been modified and executed as a single unit transferring information between the two component codes after every time step during the calculations (interaction mode).

In the non-interaction mode, the STRUCTURE code assumes the fluid to be a solid mass with the mass concentrated at the initial location of the mass center of the tank-fluid system. In other words, during the entire flexible system dynamics calculations, the fluid retains its initial position and shape. Combined with the fluid dynamic calculations (interaction mode), STRUCTURE gets the location of the mass center and the six components of moments of inertia, as input at the beginning of every time step. This would help correlate the effects of the sloshing liquid on the structure more accurately.

Similarly, in the non-interaction mode, SLOSH3D assumes that the tank undergoes only a simple rotary motion about the principal spin axis and that there is no rotation or translation in the other directions. As part of an interactive code, the instantaneous positions, velocities, and accelerations of the tank are passed on from STRUCTURE to SLOSH3D at the end of every time step, resulting in a better prediction of the sloshing motion.

This kind of interactive calculation procedure is very essential in order to investigate the fluid-structure system as one unit. It was hoped that this would help better understand the interaction between the fluid and structural elements, especially the details of unstable spin-up motions.

6.2 Logic of interaction

Figure 6.1 shows a schematic of the interaction between the computational fluid dynamics(CFD) code and the structural dynamics code. A master program calls the CFD code which calculates the fluid flow variables after one time step, typically 0.001 seconds. The inertial properties, namely, the components of the moments of inertia dyadic and the coordinates of the center of mass which have been calculated by the CFD code are passed on to the structural code. Using these data as part of its input, the structural dynamics code advances the calculations through one time step. Kinematic properties (positions, velocities, and accelerations in all the three coordinate directions) of the geometric center of the tank computed by the structural code form part of the input to the CFD calculations for the next time step.

In this fashion, the two codes are called alternatively by the master program to progress the solutions in time. Three different levels of interaction can be identified as listed below:

- *No interaction*

This is equivalent to running the two codes separately as stand alone units with no information transfer. Physically, the model is represented by Figure 6.2a.

- *One-way interaction*

In this option, the information transfer is only one way, i.e., from the CFD code to the structural dynamics code. There is no information input to the CFD code from STRUCTURE.

- *Two-way interaction*

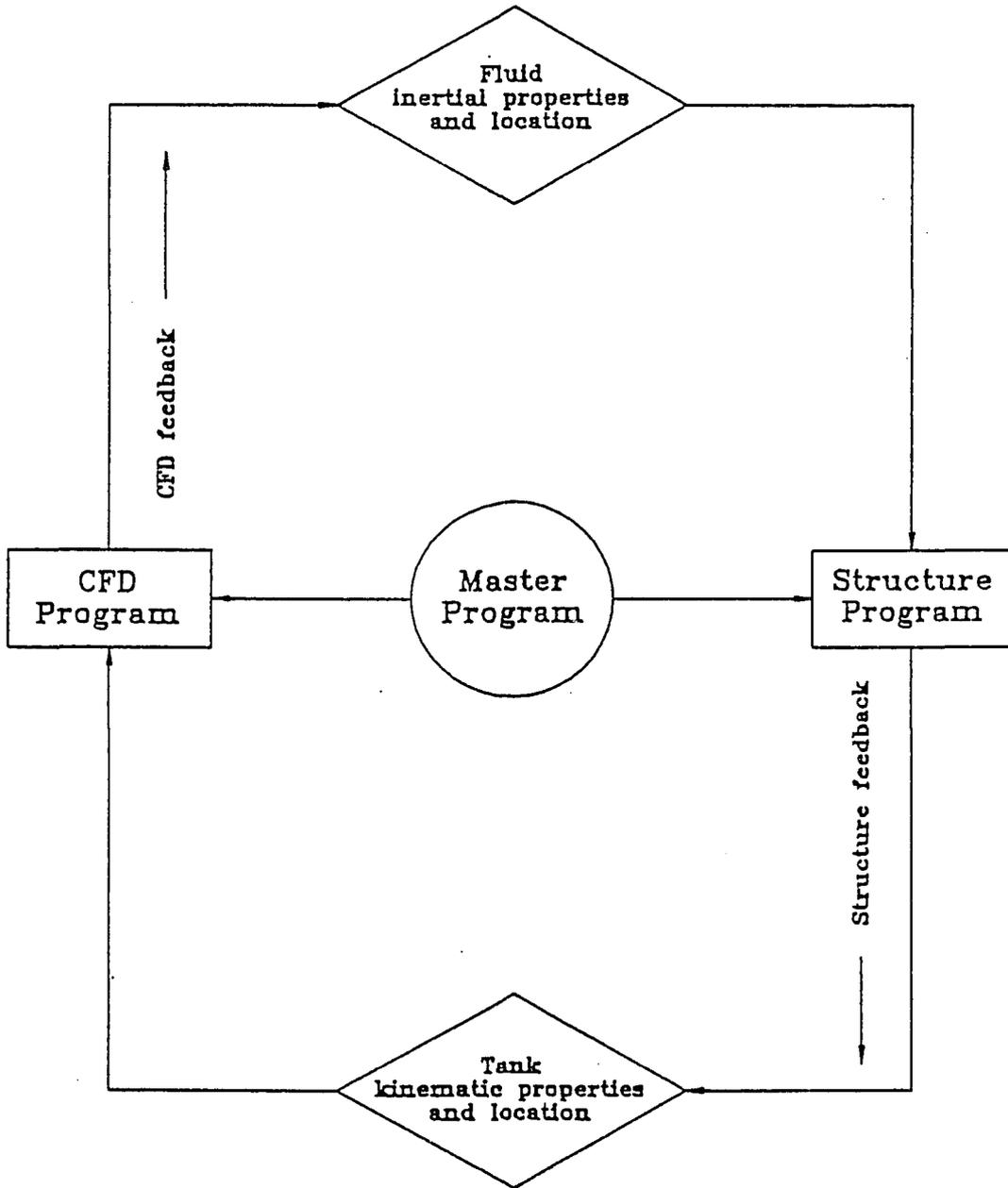


Figure 6.1: Logic of Interaction Program

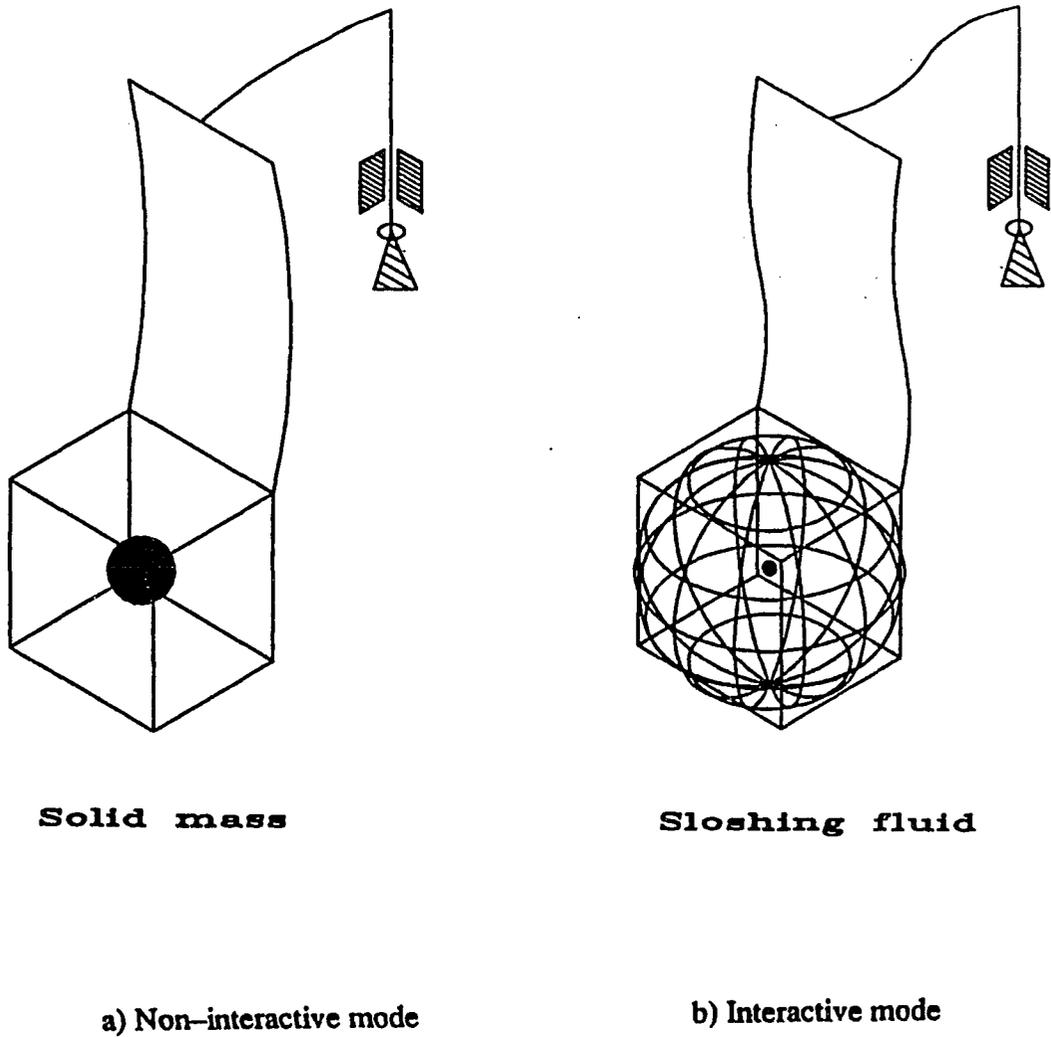


Figure 6.2: Explanation different modes of interaction

Information goes back and forth between the two codes as explained at the beginning of the section. Figure 6.2b shows the physical representation of this interaction mode.

Simple stable spin-up cases have been computed under these different levels of interaction and the initial results are discussed. Experiments have also been conducted on the satellite test-rig at Iowa State University. The numerical calculations were performed so that the results from the computations could be verified using the experimental data.

6.3 Test case and results

A stable spin-up case starting from rest and climbing to 60 RPM in 0.5 seconds according to the function:

$$\omega = \left[1 - \cos\left(\frac{t\pi}{0.5}\right) \right]$$

The rotational ω speed remains constant at 60 RPM after 0.5 seconds.

Figure 6.3 shows the tangential and radial deflections of the tank's geometric center with time as computed under all the three modes of interaction. Computed results for the transient motion of the free-surface contact points are shown in Figure 6.4. It can be seen that there is only very little difference between the deflections calculated under no-interaction(solid curve) and one-way interaction(short dashed curve). There is little difference between the results given by one-way and two-way(long dashed curve) interactions. In fact, the data for the two curves are so close that it is hard to differentiate one curve from the other on the chosen scale. From

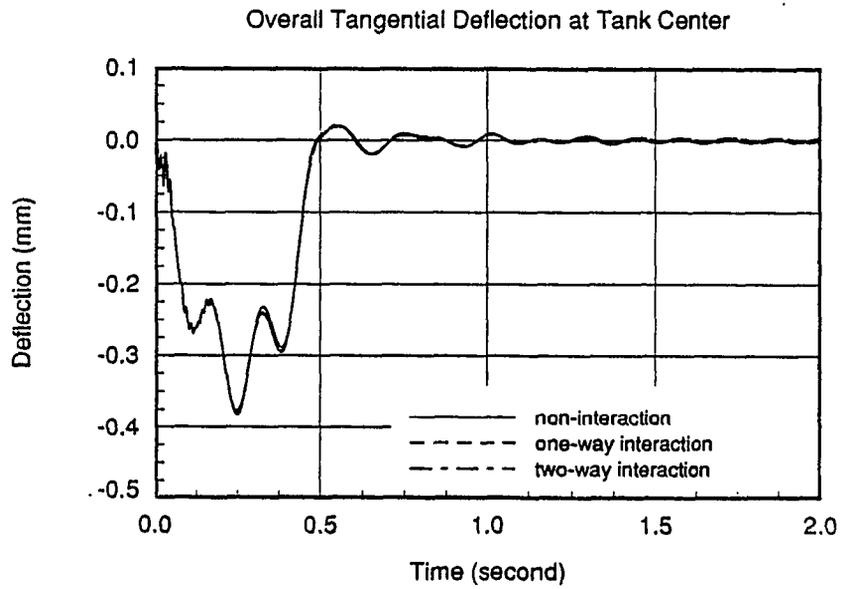
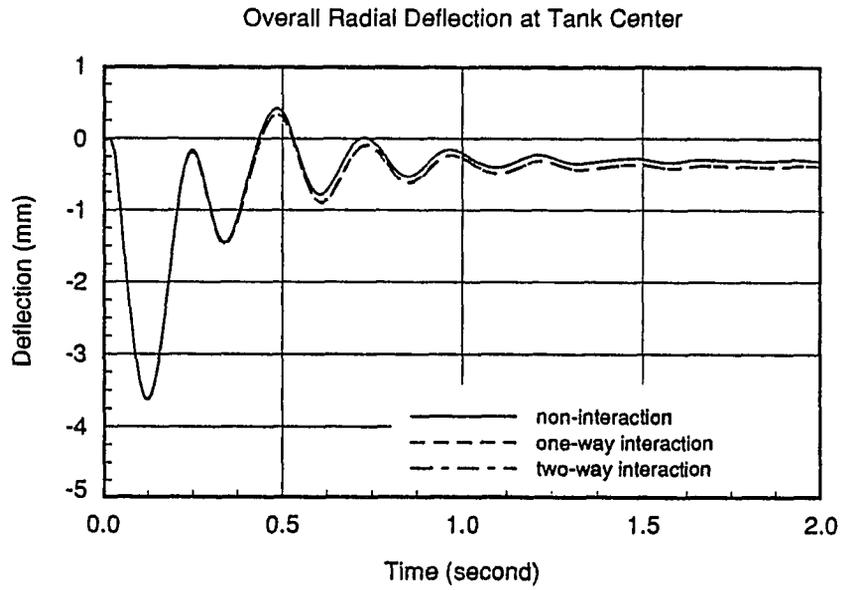


Figure 6.3: Transient tank deflection data

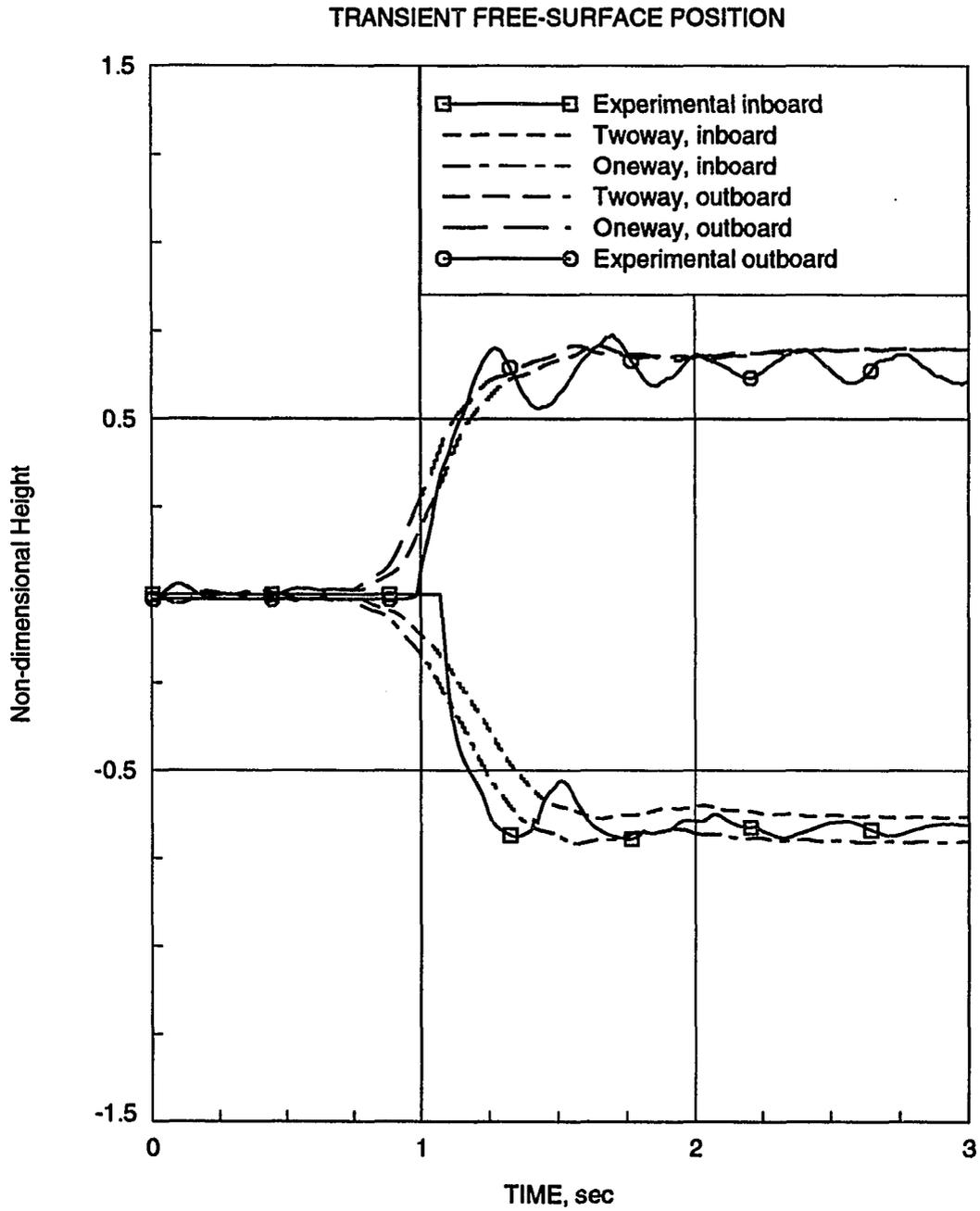


Figure 6.4: Transient free surface contact point motion

this figure, it looks like one-way interaction is sufficient to get good results for simple stable spin-up cases like the one discussed. This point needs to be studied further with more test cases before such a conclusion can be made. It should however be pointed out that the usefulness of two-way interaction should not be underestimated based on the above results. It is believed that two-way interaction results would be better and would more accurately simulate real conditions in the case of an unstable or asymmetrical spin-up similar to the situations where the collar is 'down'.

During this initial run, it was noticed that the computation intensive CFD code became too slow and time consuming under interactive operation. The non-interactive run took about 30 hours and the two-way interaction run consumed as much as 75 hours on a DECstation 5000 machine. This is one reason the interactive calculations could not be carried out beyond 0.44 seconds.

More experiments as well as numerical calculations have been performed to validate the interaction code. The mode of interaction chosen for all these cases is the two-way interaction, where both the fluid and structural codes pass information back and forth at every time step. The experiments and the corresponding computed results are discussed in the next section.

6.4 Experiments

The satellite test-rig at Iowa State University and the associated instrumentation of the rig have been well documented. A schematic of the test rig is shown in Figure 2.1 in chapter two. The upper body of the rig emulates an orbiting satellite and is mounted on an universal joint. A DC motor provides the necessary power to the system through a gear train. A collar supports the horizontal bar to which are attached

two spherical tanks half filled with glycerine. Threaded rods are used to support the spherical tanks to the horizontal cross bar. The collar could be positioned 'up' where it restrains the horizontal bar in its position and allows the spherical tanks to simply rotate about the main vertical axis. The collar could also be positioned 'down' where a more general rotation-nutation motion of the spherical tanks is possible. In this position, the horizontal bar is free to fully rotate in all three directions about the universal joint.

It is appropriate to point out the main difference between these two options in the collar position. In its 'up' position, the motion of the rig is symmetrical about its vertical axis. CFD simulation of one spherical tank and also the structural solution for one half of the system should suffice. This is not necessarily true in the situations where the collar is 'down'. Due to the enhanced freedom of motion at the universal joint, numerical simulations of both halves of the fluid-structure system are essential. This means that the motion of fluids in both tanks have to be solved at each time step and stored separately. This dramatically increases the computational as well as the memory requirements for a calculation. Owing to this, all the cases considered for this initial study belong to the collar 'up' situation.

Measurements of the following transient data could be made using the existing instrumentation:

1. Angular velocity of the rig's upper body
2. Displacements of the tank positions due to the bending of the supporting threaded rods
3. Displacement of the fluid free-surface in the tank at three different circumfer-

ential positions

A tachometer connected to the shaft in the drive train provided data on the rotational velocities. Strain gauges positioned on the cross bar generated data representing the displacement of the tank center from the main vertical axis of the upper body. Three photopotentiometers mounted on the tanks were used to obtain data on the free-surface position at three different locations, namely the 'inboard', 'outboard' and the 'lagging side'. The inboard end refers to the point closest to the main axis, the outboard end the point farthest from the main axis of symmetry. The lagging side denotes the end of the spherical tank that geometrically lags the rest of the sphere in the rotational sense.

Calibration was performed for all three types of data so that the raw voltages measured during the experiments could be converted to useful data for interpretation. An IBM PS/2 along with a National Instruments Data Acquisition card was used to acquire and process the data through software written in Basic. A nominal scan rate of 100 scans per second was used for the experimental runs. The experiments were performed with the rig initially at rest and gradually gaining speed until achieving the final specified rotational speed. Spin-ups were performed to three different final rotational speeds: 30, 45 and 60 rpm. The results were qualitatively very similar among the runs at different speeds. Hence, only some of the results are shown; the results from the 30 and 60 rpm trials are shown in this thesis.

6.5 Numerical Simulation

Two of the experimental runs described above were taken for comparison with results from computer simulations. One of the main inputs to the numerical compu-

tations is the rotational speed of the test rig as a function of time. The data obtained from the experiment was smoothed to remove measurement noise before being input to the numerical computations. This was done to minimize possible numerical instabilities due to the oscillatory data. Figure 6.5 shows a sample of the data that was experimentally measured and the corresponding smoothed data that was input to the computations.

The computations were performed on a Cray Y-MP and took approximately 200 minutes of CPU time for each spin-up calculation. The calculations involved marching the solution in time for 3000 timesteps (each equal to 0.001 seconds) corresponding to a total time of 3.0 seconds of the actual spin-up experiment.

6.5.1 Results and Discussions

Two computer runs were attempted: one for the final spin-up speed of 60 RPM and the other for 30 RPM. As the nature of the computed results were similar in both the cases, only one of them (corresponding to a final spin-up speed of 60 RPM) is discussed here. Figure 6.6 shows the comparison of the free surface positions (inboard and outboard) between the experimental and computed results for the 60 RPM case. It can be seen that the transients compare reasonably well for the inboard end. The outboard data do not compare as well, and the difference in the final outboard free-surface position between the experiment and the numerical computation is due to several factors. The most important of these being the experimental uncertainties in measuring the free-surface position, the uncertainty in the amount of liquid in the tank (computations assumed a fill ratio of 0.5), and computation of a tank angle that was too small due to the rigid cross-arm assumption, which is discussed below

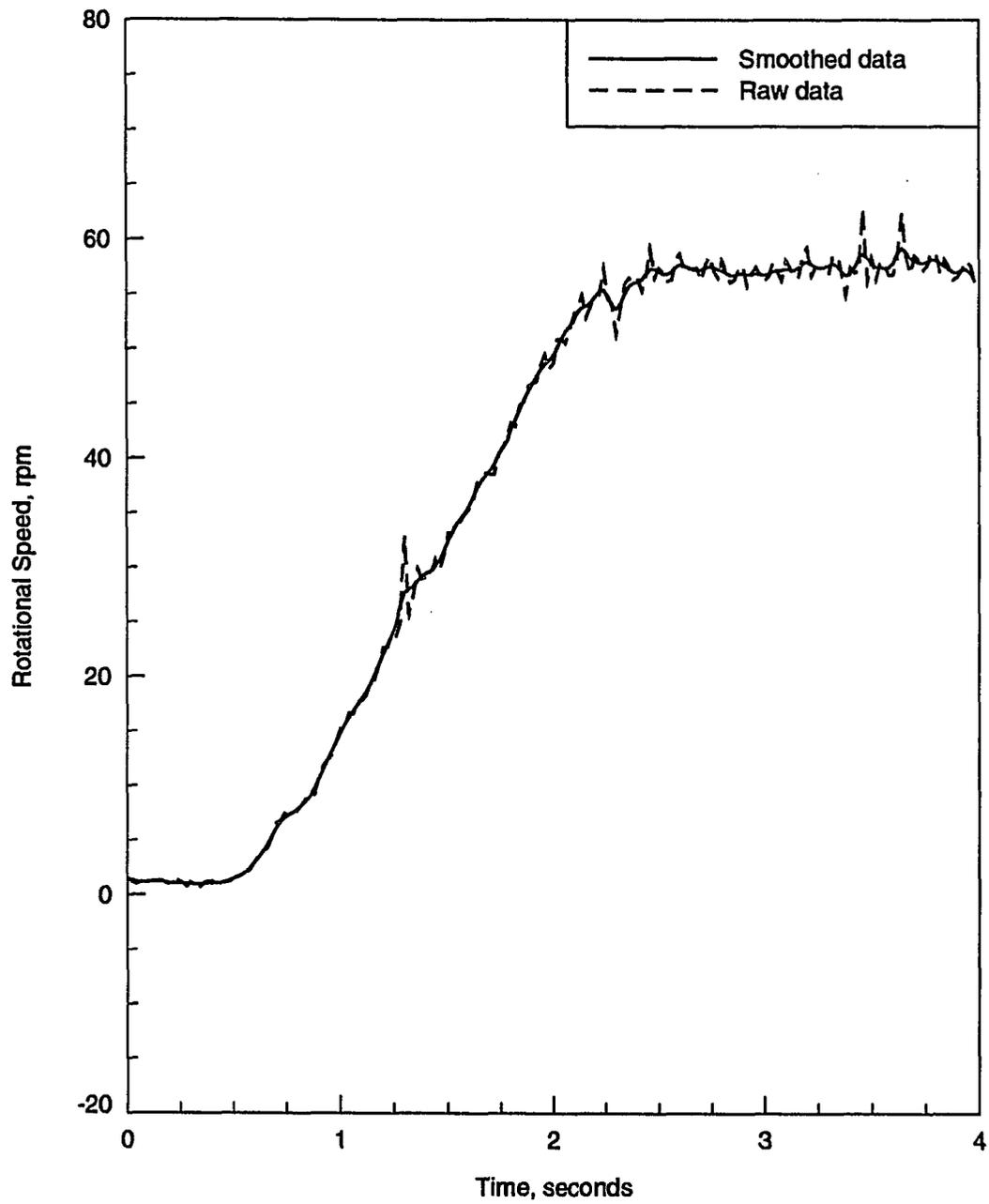


Figure 6.5: Comparison of raw and smoothed transient angular velocity data.

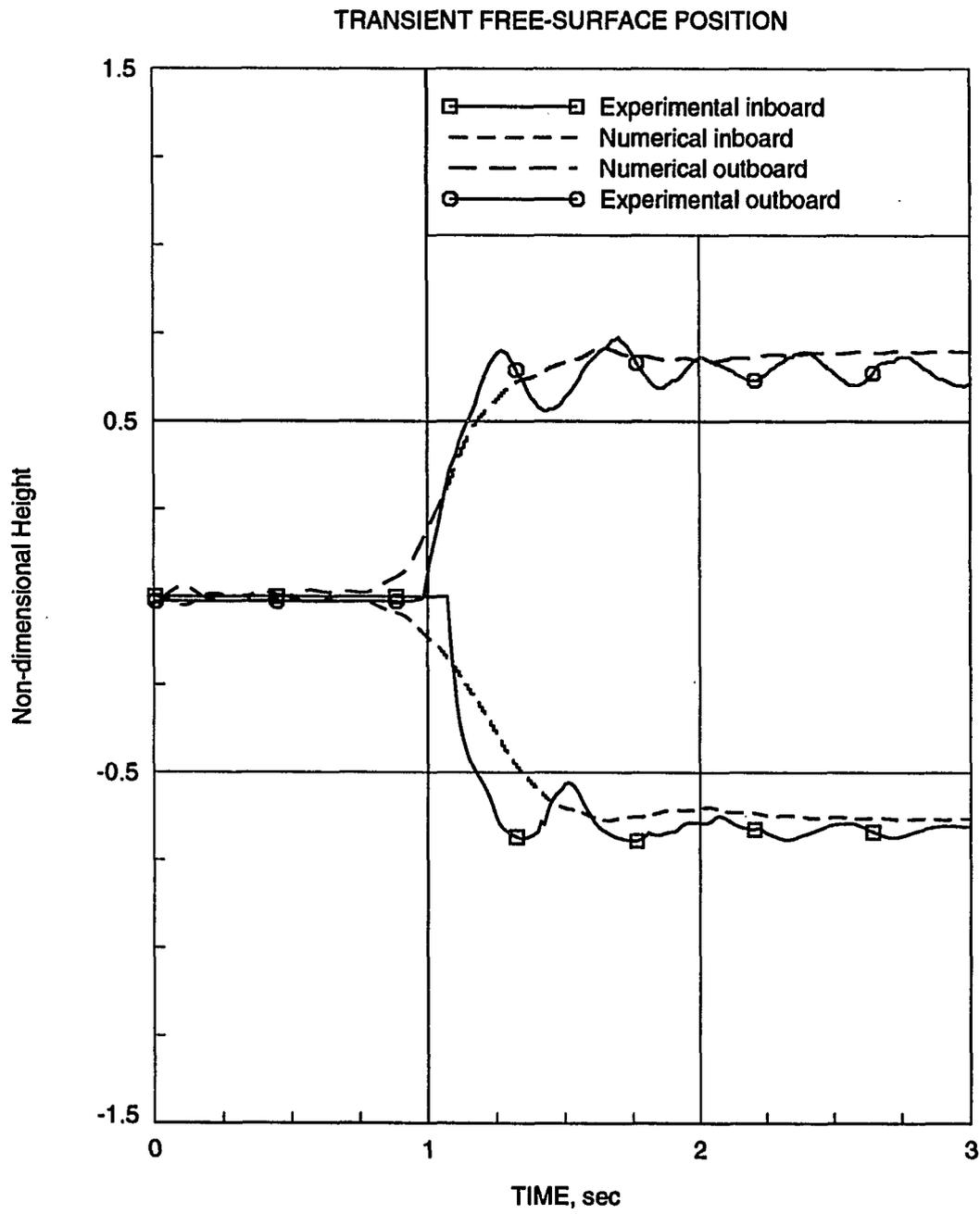


Figure 6.6: Comparison of numerical and experimental free-surface transient data

in more detail. The numerical results also seem to be more smooth compared to the experimental data. This is attributed to the relatively coarse grid used in the computations ($41 \times 11 \times 11$).

Figure 6.7 shows the comparison between the experimental and numerically computed radial positions of the tank center. It can be seen that the comparison is reasonably good and the deviations are explained below. The final radial position of the tank seems to have been predicted well by the code. Probably the most significant source of discrepancy in the radial deflection values is that the STRUCTURE code assumed a perfectly rigid cross arm on the upper body of the test rig. In reality, a small deflection in this bar would result in a relatively large deflection in the position of tank.

The rigid cross-arm assumption could also have lead to the higher frequency of oscillation in the computed case, which can be seen in Figure 6.7. The transient oscillations in the first one second of the computed results is due to the initial condition used and the absence of viscous damping (of the inherent oscillations introduced by the numerical integration procedure). The numerical computations assume that, at time = 0.0 seconds, the tank is at its vertical position without any static load and is suddenly subjected to the load due to the fluid mass at time greater than zero. This is thought to be the main cause of the oscillations seen in the computed results.

It was realized from the first set of trials that the initial conditions that existed during the experiments were not the same as the 'STRUCTURE' code assumed. As it was more difficult to modify the code to incorporate the actual experimental initial conditions into the calculations, the experiments were instead repeated to more realistically duplicate the conditions used by the computational procedure. The

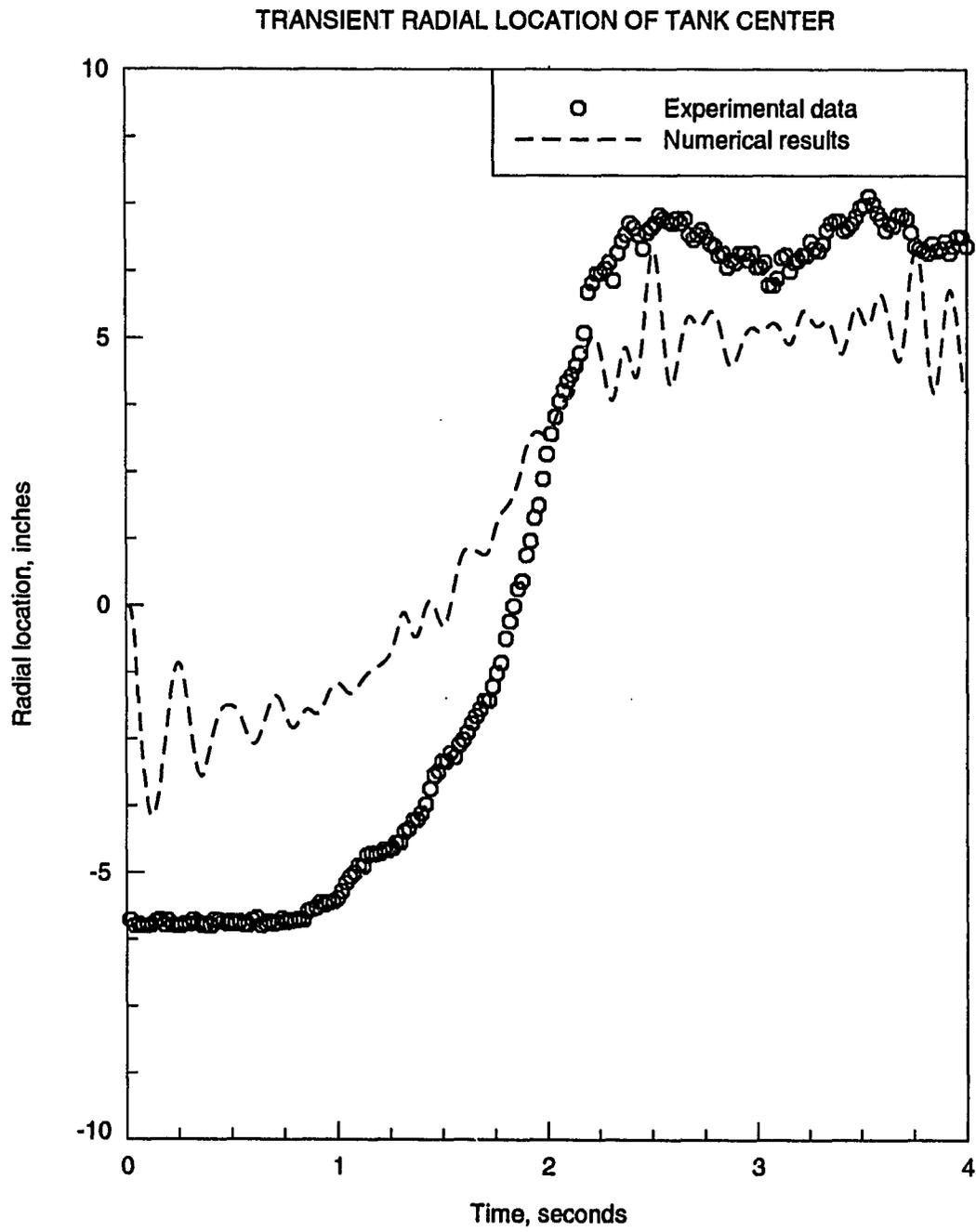


Figure 6.7: Comparison of numerical and experimental data on tank radial location

condition under question relates to the initial displacements of the tank center. The STRUCTURE code assumes that the threaded rods supporting the spherical tank containing the fluid are vertical. In reality, these rods are bent radially inwards due to the weight of the tank and fluid acting on the rods. The experiments were rerun so that the tanks were initially held apart by a metal bar of suitable length so that the rods supporting the spherical tanks were indeed vertical. This restraining bar was suddenly removed and the spin-up initiated simultaneously to match the initial conditions used in the calculations. This change in the experimental strategy resulted in the observation of oscillatory displacement data similar to those shown in the numerical results. Typical comparisons of the free-surface positions and the radial location of the tank center for the revised experimental procedures are shown in Figures 6.8 and 6.9 respectively.

6.6 Possible Improvements

Given sufficient time and resources, the experimental and computational results for the spin-up case could be brought into even closer agreement. Improvements which could be implemented include photovoltaic fluid level sensors, which would potentially be calibrated more reliably over a broader range; inclusion of a flexible crossbar in the STRUCTURE code; and a finer grid size for computing high-frequency fluid oscillations in the SLOSH3D code.

TRANSIENT FREE-SURFACE POSITION

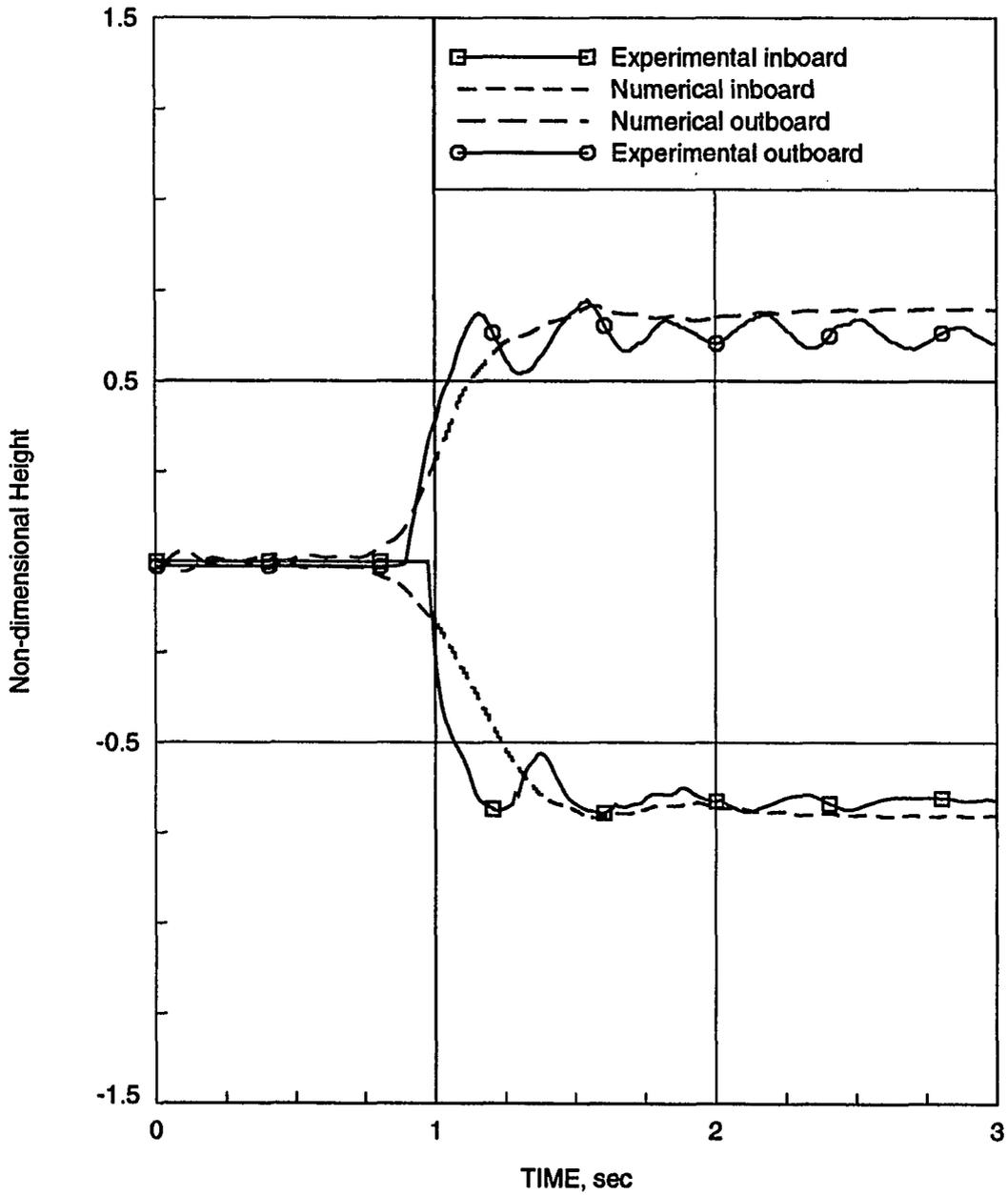


Figure 6.8: Comparison of numerical and experimental free-surface transient data, revised experiments

TRANSIENT RADIAL LOCATION OF TANK CENTER

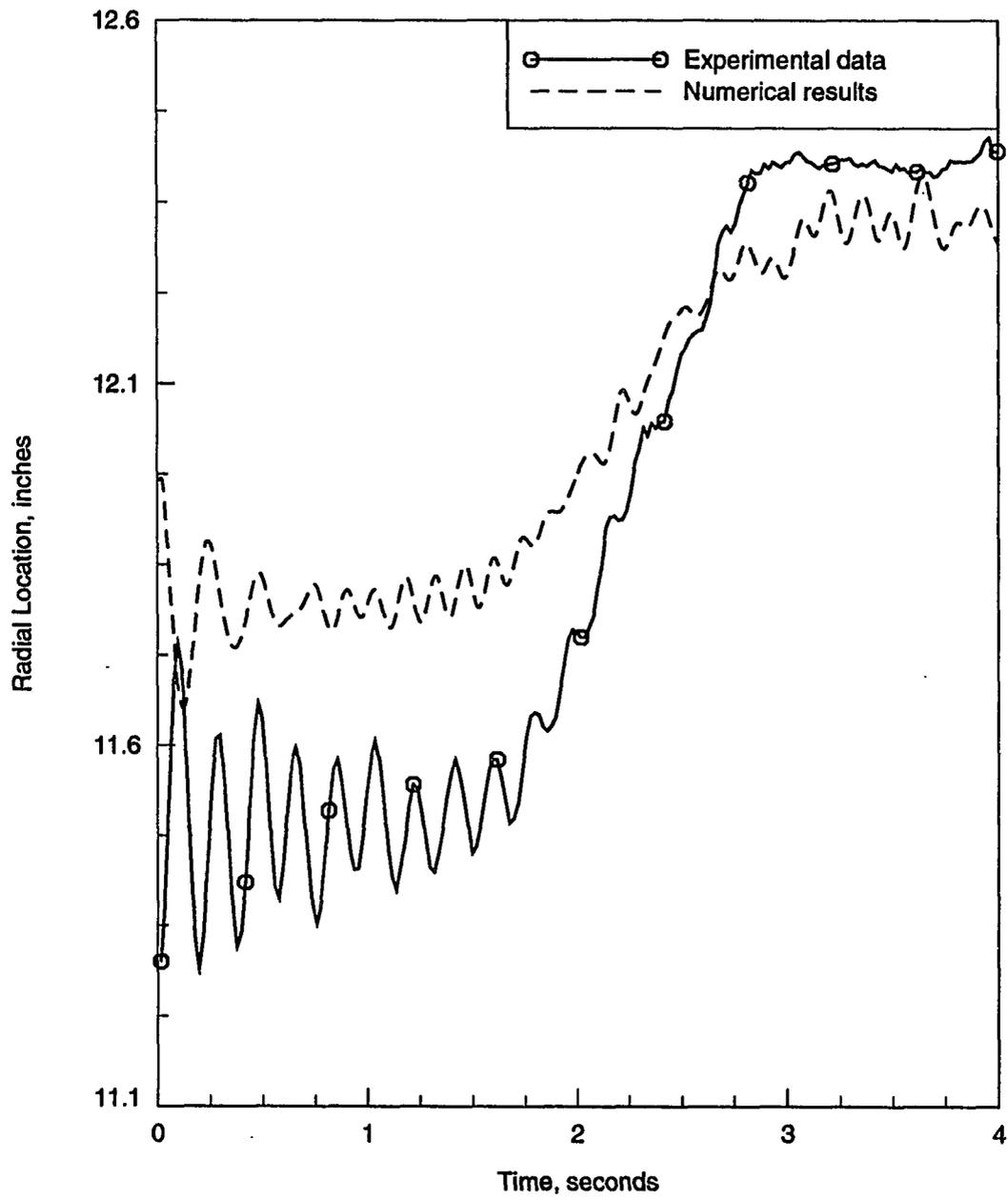


Figure 6.9: Comparison of numerical and experimental data on tank radial location, revised experiments

7. CONCLUDING REMARKS

7.1 Conclusions

Unsteady, three-dimensional incompressible flows with free-surface(s) have been successfully computed under a variety of situations. These include the sloshing motion of liquids in partially filled spherical and cylindrical containers as well as the flow of a three-dimensional broken dam.

Three different free-surface methods have been tried and it was found that the last of these, namely method 'C' showed the greatest promise. The transient results from this procedure was found to be in good agreement with published experimental data. During this study, it was found that method 'C' was superior to method 'B' which in turn produced more accurate results than method 'A'. It is recommended that the method 'C' be used for future calculations using the surface-fitting procedure.

The zero normal slope for the contact line gave best results for the free-surface computations. This method is favored over extrapolation or other slip conditions. Benefits from this boundary condition included a 'better' interior grid (and hence enhanced overall stability for the computations) and a more accurate free-surface transience.

Preliminary results show that the formulation is capable of computing flows under microgravity environments where the forces due to surface tension become sig-

nificant. The feasibility of combining the liquid sloshing CFD computations with a finite element structural dynamics program (fluid-structure interaction) has also been demonstrated. The initial results from this effort were found to be in reasonable agreement with experimental data. The computational efficiency of the algorithm and the total computation as such has been tremendously increased owing to the vectorization and other optimization efforts. The calculation procedure has demonstrated in many ways that it possesses the capability to predict complex three-dimensional free-surface flows.

7.2 Suggestions for further research

Though every effort has been made to keep all aspects of the calculations as complete as possible, certain areas merit additional attention. These points are discussed below.

1. The decision on the appropriate instance when a new grid generation is required can be made more automatic. Presently, this is simply done after every predetermined number of subiterations in a physical time step.
2. More microgravity calculations can be performed to validate the procedure. Different container shapes and varying gravity levels within a computational run (to simulate the effect of gravity jitters) are certain possibilities.
3. It would be highly desirable to perform an asymmetric calculation with the fluid-structure interaction code. This could not be done in the current research due to time and other resource restrictions.

BIBLIOGRAPHY

- [1] Chen, C. L., Ramakrishnan, S., and Szema, K. Y. (1992). "Multizonal Navier-Stokes solutions for the multibody space shuttle configuration," *Journal of Spacecraft and Rockets*, Vol. 29, Jan./Feb. '92, pp. 129-37.
- [2] Chiu, I. T. (1990). "On Computations of the Integrated Space Shuttle Flowfield Using Overset Grids." Ph.D. Dissertation. Iowa State University, Ames, Iowa.
- [3] Abramson, H.N. (1966). "The Dynamic Behavior of Liquids in Moving Containers," Southwest Research Institute, NASA SP-106.
- [4] Taylor, T. W., and Yamaguchi, T. (1994). "Three-dimensional simulation of Blood Flow in Abdominal Aortic Aneurysm - Steady and Unsteady Flow Cases," *Journal of Biomechanical Engineering*, Vol. 116, pp. 89-97.
- [5] Cooper, R.M. (1960). "Dynamics of Liquids in Moving Containers," *ARS Journal*, Vol. 30, No. 8, pp. 725-7
- [6] Peterson, L. D., Crawley, E. F. and Hansman, R. J. (1989). "Nonlinear Fluid Slosh Coupled to the Dynamics of a Spacecraft." *AIAA Journal*, Vol. 27, No. 9, pp. 1230-1240.

- [7] Kana, D. D. "Validated Spherical Pendulum Model for Rotary Liquid Slosh." (1989). *Journal of Spacecraft and Rockets*, Vol. 26, No. 3, pp. 188-195.
- [8] Patankar, S. V. (1981). "Numerical Heat Transfer and Fluid Flow," Hemisphere Publishing Corp., New York.
- [9] Roache, P. (1972). "Computational Fluid Dynamics." Hermosa Publishers, Albuquerque, New Mexico.
- [10] Osswald, G. A., Ghia, K. N. and Ghia, U. (1987). "A Direct Algorithm for Solution of Incompressible Three-Dimensional Unsteady Navier-Stokes Equations." AIAA-87-1139.
- [11] Pan, D. and Chakravarthy, S. (1989). "Unified Formulation for Incompressible Flows." AIAA-89-0122.
- [12] Gegg, S. "A Dual Potential Formulation of the Navier-Stokes Equations." Ph.D. Dissertation, Iowa State University, Ames, Iowa, 1989.
- [13] 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.
- [14] 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.

- [15] 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.
- [16] 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.
- [17] 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.
- [18] Chen, K-H., and Pletcher, R.H. (1991) "Simulation of Three-Dimensional Liquid Sloshing Flows Using a Strongly Implicit Calculation Procedure," AIAA 91-1661.
- [19] Peterson, L. D., Crawley, E. F. and Hansman, R. J. (1989). "Nonlinear Fluid Slosh Coupled to the Dynamics of a Spacecraft," *AIAA Journal*, Vol. 27, No. 9, pp. 1230-1240.
- [20] Chorin, A.J., (1967). "A Numerical Method for Solving Incompressible Viscous Flow Problems," *J. Comp. Phys.*, Vol. 2, pp. 12-26.
- [21] Merkle, C. L., and Athavale, M. (1987). "Time-Accurate Unsteady Incompressible Flow Algorithm based on Artificial Compressibility," AIAA 87-1137, 1987.
- [22] Rogers, S. E., and Kwak, D., (1988). "An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations," AIAA 88-2583, 1988.

- [23] Rosenfeld, M., and Kwak, D., (1993). "Multigrid Acceleration of a Fractional-Step Solver in Generalized Curvilinear Coordinate Systems," *AIAA Journal*, Vol. 31, Oct. '93, pp. 1792-800.
- [24] Stone, H.L., (1968). "Iterative Solution of Implicit Approximations of Multidimensional Partial Equations," *SIAM J. Numer. Anal.*, Vol. 5, pp. 530-558.
- [25] Zedan, M., and Schneider, G. E. (1983). "A Three-Dimensional Modified Strongly Implicit Procedure for Heat Conduction," *AIAA Journal*, Vol. 21, No. 2, pp. 295-303.
- [26] Rubin, S. G., and Khosla, P. G. (1981). "Navier-Stokes Calculations with a Coupled Strongly Implicit Method - I : Finite Difference Solutions," *Computers and Fluids*, Vol. 9, pp. 163-180.
- [27] Ghia, U., Ghia, K. N. and Shin, C. T. (1982). "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method," *Journal of Computational Physics*, pp. 387-411.
- [28] Phinney, R. A., Odom, R. I., and Fryer, G. J. (1987). "Rapid generation of synthetic seismograms in layered media by vectorization of the algorithm," *Bulletin of the Seismological Society of America*, Vol. 77, pp. 2218-26.
- [29] Majumdar, S. R., and Zhu, W. J. (1992). "Three-dimensional Finite Volume Method for Incompressible Flows with Complex Boundaries," *Journal of Fluids Engineering*, Vol. 114, pp. 496-503.

- [30] 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.
- [31] 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.
- [32] 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.
- [33] Breit, S. R., Coney, W. C., and Dickinson, A. L. (1992). "Computing Boundary Forces due to Unsteady, Inviscid, Incompressible Flow," *AIAA Journal*, Vol. 30, pp. 592-600.
- [34] 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.
- [35] Concus, P., Finn, R. and Weislogel, M. (1992). "Drop-tower Experiments for Capillary Surfaces in an Exotic Container," *AIAA Journal*, 30, pp 134-7.
- [36] Kothe, D. B., and Mjolsness, R. C. (1992). "RIPPLE: A New Model for Incompressible Flows with Free-Surfaces," *AIAA Journal*, Vol. 30, pp. 2694-700.
- [37] Xu, J., and Baumgarten, J. R. (1992). "A sequential Implicit-Explicit Integration Method in Solving Nonlinear Differential Equations from Flexible Sys-

- tem Modeling," *Flexible Mechanisms, Dynamics, and Analysis, ASME*, Vol. 47, pp. 561-566.
- [38] Xu, J., and Baumgarten, J. R. (1992). "Modeling of Multibody Flexible Articulated Structures with Mutually Coupled Motions: Parts I and II," *Flexible Mechanisms, Dynamics, and Analysis, ASME*, Vol. 47, pp. 411-426.
- [39] Paidoussis, M. P. (1993). "1992 Calvin Rice Lecture: Some Curiosity-Driven Research in Fluid-Structure Interaction and its Current Applications," *Journal of Pressure Vessel Technology*, Vol. 115, pp 14-20.
- [40] Kim, I., Elghobashi, S., and Sirignano, W. A. (1993). "Three-dimensional Flow over Two Spheres Placed Side by Side," *Journal of Fluid Mechanics*, Vol. 246, pp. 465-88.
- [41] Sauve, R. G., Morandin, G. D., and Nadeau, E. (1993). "Impact Simulation of Liquid-Filled Containers Including Fluid-Structure Interaction," *Journal of Pressure Vessel Technology*, Vol. 115, pp 79-85.
- [42] Kelecy, F. J., (1993) "Numerical simulation of two and three-dimensional viscous free surface flows in partially-filled containers using a surface capturing approach." Ph.D. dissertation, Iowa State University, Ames, Iowa, 1993.
- [43] Huh, C., and Scriven, L. E. (1971). "Hydrodynamic Model of Steady Movement of a Solid / Liquid Fluid Contact Line," *Journal of Colloidal Interface Science*, Vol. 35, pp. 85-101.

- [44] Kistler, S. F., and Scriven, L. E. (1983). "Coating Flows, in Computational Analysis of Polymer Processing," J. R. A. Pearson and S. M. Richards Eds., Applied Science, London, pp. 243-299.
- [45] E. B. Dussan V., (1976) "The moving contact line: the slip boundary condition," *J. Fluid Mech.*, Vol. 77, pp. 665-684.
- [46] E. B. Dussan V. and Stephen H. Davis, (1974) "On the motion of a fluid-fluid interface along a solid surface," *J. Fluid Mech.*, Vol. 65, pp. 71-95.
- [47] S. H. Davis, (1983). "Contact-Line Problems in Fluid Mechanics," *Journal of Applied Mechanics*, Vol 50. pp 977-981.
- [48] 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.
- [49] Silliman, W. J. and Scriven, L. E. (1980). "Separating flow near a static contact line; Slip at a wall and shape of a free-surface," *J. Comput. Phys.*, Vol. 34 pp. 287-313.
- [50] Anderson, D.A., Tannehill, J.C., and Pletcher, R.H. (1984). "Computational Fluid Mechanics and Heat Transfer," McGraw-Hill Book Co., New York.
- [51] Chen, K-H., Kelecy, F.J., Pletcher, R.H., (1992). "A Numerical and Experimental Study of Three-Dimensional Liquid Sloshing in a Rotating Spherical Container," AIAA 92-0829.

- [52] 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.
- [53] Penney, W. G., and Thornhill, C. K. (1952). "The Dispersion Under Gravity, of a Column of Fluid Supported on a Rigid Horizontal Plane," *Philosophical Transactions of the Royal Society of London*, Ser. A, Vol. 244, pp. 285-311.
- [54] Penney, W. G. and Price, A. T. (1952). "Finite Periodic Stationary Waves in a Perfect Liquid," *Philosophical Transactions of the Royal Society of London*, Ser. A, Vol. 244, pp. 254-284.
- [55] Chandrasekhar, S. (1965). "The Stability of a Rotating Liquid Drop," *Proceedings of the Royal Society of London, A*, Vol. 286.
- [56] 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.
- [57] 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.
- [58] Lamb, H. (1945). "Hydrodynamics," Sixth ed., Dover publ., New York.
- [59] Homicz, G. F., and Gerber, N. (1987). "Numerical Model for Fluid Spin-Up from Rest in a Partially Filled Cylinder," *The Journal of Fluids Engineering*, Vol. 109, pp. 194-197.

- [60] Wedemeyer, E. H. (1964). "The Unsteady Flow Within a Spinning Cylinder," *Journal of Fluid Mechanics*, Vol. 20, part 3, pp. 383-399.
- [61] Tu, Y., and Thompson, J. F. (1991). "Three-dimensional Solution-adaptive Grid Generation on Composite Configurations," *AIAA Journal*, Vol. 29, pp. 2025-2026.
- [62] Kim, H. J., and Thompson, J. F. (1990). "Three-dimensional Adaptive Grid Generation on a Composite Block Grid," *AIAA Journal*, Vol. 28, pp. 470-477.
- [63] Hindman, R. G. (1982). "Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors," *AIAA Journal*, Vol. 20, pp. 1359-1367.
- [64] Thompson, J. F., and Mastin, C. W. (1985). "Order of Difference Expressions in Curvilinear Coordinate Systems," *Journal of Fluids Engineering*, Vol. 107, pp. 241-250.
- [65] Thompson, J. F. (1985). "Numerical Grid Generation: foundations and applications," Elsevier Science Pub. Co., New York.
- [66] Noack, R. W., and Anderson, D. A. (1990). "Solution-Adaptive Grid Generation Using Parabolic Partial Differential Equations," *AIAA Journal*, Vol. 28, pp. 1016-1023.
- [67] Leslie, F.W., (1985) "Measurement of Rotating Bubble Shapes in a Low-Gravity Environment," *Journal of Fluid Mechanics*, Vol. 161, 1985, pp. 269-279.

- [68] Hung, R.J., and Leslie, F.W., (1988). "Bubble Shapes in Liquid Filled Rotating Container Under Low Gravity," *Journal of Spacecraft and Rockets*, Vol. 25, pp. 70-74.
- [69] Hung, R.J., Tsao, Y.D., Hong, B.B., and Leslie, F.W., (1988). "Time-Dependent Dynamical Behavior of Surface Tension on Rotating Fluids Under Microgravity Environment," *Advances in Space Research*, Vol. 8., pp. 205-213.
- [70] Hung, R.J., Tsao, Y.D., Hong, B.B., and Leslie, F.W., (1989). "Bubble Behaviors in a Slowly Rotating Helium Dewar in Gravity Probe-B Spacecraft Environment," *Journal of Spacecraft and Rockets*, Vol. 26, pp. 167-172.
- [71] White, F. M. (1986). "Fluid Mechanics," McGraw-Hill Book Co., New York.
- [72] Munson, B. R., Young, D. F., and Okiishi, T. H. (1986). "Fundamentals of Fluid Mechanics," John Wiley and Sons, New York.

APPENDIX A. VECTOR COMPONENTS

The unit normal and tangent vectors are given by the following expressions:

1. Wall

In the case of the liquid sloshing problems, the wall of the container always coincided with the $z_2 = \text{constant}$ surface. On the other hand, the walls for the broken dam calculations, coincided with $z_i = \text{constant}$ with 'i' running from 1 to 3. The unit normal vectors in this situation are:

$$\vec{n} = - \left[\frac{\nabla z_i}{\|\nabla z_i\|} \right]_{wall}$$

As ∇z_i is given by $\hat{e}_{3j}\eta_{i,j}$, the normal can be expressed as:

$$\vec{n} = - \left[\frac{\hat{e}_{31}\eta_{i,1} + \hat{e}_{32}\eta_{i,2} + \hat{e}_{33}\eta_{i,3}}{\sqrt{\eta_{i,1}^2 + \eta_{i,2}^2 + \eta_{i,3}^2}} \right]_{wall}$$

That is,

$$n_{i,j} = - \left[\frac{\eta_{i,j}}{\sqrt{\eta_{i,k}^2}} \right]_{wall}$$

where 'i' corresponds to the index given by the wall expression $z_i = \text{constant}$ and the 'j' stands for the component in the x_{3j} direction. Unit tangent vectors at the wall are not required in the formulation.

2. Free-surface

Free-surface in all the problems considered for this research coincide with one $z_i = \text{constant}$ surface. Hence, the normal vectors at the free-surface are given by:

$$n_i = - \left[\frac{\eta_{3,i}}{\sqrt{\eta_{3,k}^2}} \right]_{\text{free-surface}}$$

The two unit tangent vectors follow from unit length requirements and orthogonality conditions. They are given by:

- Free-surface is given by $z_3 = \text{constant}$

$$\vec{\tau}_1 = \frac{\hat{e}_{31} - \hat{e}_{33}\alpha}{\sqrt{1 + \alpha^2}}$$

$$\vec{\tau}_2 = \frac{-\hat{e}_{31}\alpha\beta + \hat{e}_{32}(1 + \alpha^2) - \hat{e}_{33}\beta}{\sqrt{1 + \alpha^2 + \beta^2}\sqrt{1 + \alpha^2}}$$

where $\alpha = \frac{n_1}{n_3}$ and $\beta = \frac{n_2}{n_3}$.

- Free-surface is given by $z_2 = \text{constant}$

$$\vec{\tau}_1 = \frac{\hat{e}_{31} - \hat{e}_{32}\alpha}{\sqrt{1 + \alpha^2}}$$

$$\vec{r}_2 = \frac{-\hat{e}_{31}\alpha\beta - \hat{e}_{32}\beta + \hat{e}_{33}(1 + \alpha^2)\beta}{\sqrt{1 + \alpha^2 + \beta^2}\sqrt{1 + \alpha^2}}$$

where $\alpha = \frac{n_1}{n_2}$ and $\beta = \frac{n_3}{n_2}$.

APPENDIX B. LISTING OF CSIP FORTRAN ROUTINES

```

C *****
C *
C * Subroutine: CSIP3D
C * Author: S. Babu
C * Version: 1.0
C * Date: December 1991
C *
C * 3D block version of Stone's SIP algorithm.
C *
C *****

SUBROUTINE CSIP3D(il,iu,jl,ju,kl,ku,ae,aw,an,as,at,ab,ap,rh,alph)

INCLUDE 'parameter.inc'

DIMENSION ae(0:l+1,0:m+1,0:n+1,no,no),aw(0:l+1,0:m+1,0:n+1,no,no),
1          an(0:l+1,0:m+1,0:n+1,no,no),as(0:l+1,0:m+1,0:n+1,no,no),
2          at(0:l+1,0:m+1,0:n+1,no,no),ab(0:l+1,0:m+1,0:n+1,no,no),
2          ap(0:l+1,0:m+1,0:n+1,no,no),rh(1,m,n,no)
DIMENSION t(0:l+1,0:m+1,0:n+1,no)
DIMENSION a(no,no),d(no,no),f(no,no),g(no,no)
DIMENSION h(no,no),r(no,no),v(no,no)

C -----
C *** Initialization
C -----

DO 1000 k = kl-1,ku+1
  DO 1000 j = jl-1,ju+1
    DO 1000 j1 = 1,no
      t(il-1,j,k,j1) = 0.0d0

```

```

      t(iu+1,j,k,j1) = 0.0d0
1000 CONTINUE

```

```

      DO 2000 k = kl-1,ku+1
        DO 2000 i = il-1,iu+1
          DO 2000 j1 = 1,no
            t(i,jl-1,k,j1) = 0.0d0
            t(i,ju+1,k,j1) = 0.0d0
2000 CONTINUE

```

```

      DO 3000 j = jl-1,ju+1
        DO 3000 i = il-1,iu+1
          DO 3000 j1 = 1,no
            t(i,j,kl-1,j1) = 0.0d0
            t(i,j,ku+1,j1) = 0.0d0
3000 CONTINUE

```

```

c -----
c *** BEGIN LOOP
c -----

```

```

      DO 4000 k = kl,ku
        DO 4000 j = jl,ju
          DO 4000 i = il,iu

```

```

c -----
c *** Coefficients a,d,f
c -----

```

```

          DO 100 j1 = 1,no
            DO 100 i1 = 1,no
              a(i1,j1) = ab(i,j,k,i1,j1)
              d(i1,j1) = as(i,j,k,i1,j1)
              f(i1,j1) = aw(i,j,k,i1,j1)
100      CONTINUE

```

```

c -----
c *** Coefficient g
c -----

```

```

DO 250 j1 = 1,no
  DO 250 i1 = 1,no
    g(i1,j1) = ap(i,j,k,i1,j1)
    DO 300 k1 = 1,no
      g(i1,j1) = g(i1,j1)
1         -a(i1,k1)*( at(i,j,k-1,k1,j1)
2         -alph*ae(i,j,k-1,k1,j1)-alph*an(i,j,k-1,k1,j1))
3         -d(i1,k1)*(an(i,j-1,k,k1,j1)
4         -alph*ae(i,j-1,k,k1,j1)-alph*at(i,j-1,k,k1,j1))
5         -f(i1,k1)*(ae(i-1,j,k,k1,j1)
6         -alph*an(i-1,j,k,k1,j1)-alph*at(i-1,j,k,k1,j1))
300      CONTINUE
250     CONTINUE

```

```

c -----
c *** Compute inverse of g
c -----

```

CALL INV(g)

```

c -----
c *** Coefficients h,r,v
c -----

```

```

DO 350 j1 = 1,no
  DO 350 i1 = 1,no
    h(i1,j1) = 0.0d0
    r(i1,j1) = 0.0d0
    v(i1,j1) = 0.0d0
    DO 370 k1 = 1,no
      h(i1,j1) = h(i1,j1)+g(i1,k1)*ae(i,j,k,k1,j1)
      r(i1,j1) = r(i1,j1)+g(i1,k1)*an(i,j,k,k1,j1)
      v(i1,j1) = v(i1,j1)+g(i1,k1)*at(i,j,k,k1,j1)
370     CONTINUE
350     CONTINUE

```

```

c -----
c *** Save h,r,v
c -----

```

```

DO 380 j1 = 1,no
  DO 380 i1 = 1,no
    ae(i,j,k,i1,j1) = h(i1,j1)
    an(i,j,k,i1,j1) = r(i1,j1)
    at(i,j,k,i1,j1) = v(i1,j1)
380 CONTINUE

```

```

c -----
c *** Forward sweep
c -----

```

```

DO 400 i1 = 1,no
  t(i,j,k,i1) = 0.0d0
  DO 420 k1 = 1,no
    t(i,j,k,i1) = t(i,j,k,i1)+g(i1,k1)*rh(i,j,k,k1)
    DO 440 k2 = 1,no
      t(i,j,k,i1) = t(i,j,k,i1)
1      -g(i1,k1)*a(k1,k2)*t(i,j,k-1,k2)
2      -g(i1,k1)*d(k1,k2)*t(i,j-1,k,k2)
3      -g(i1,k1)*f(k1,k2)*t(i-1,j,k,k2)
440 CONTINUE
420 CONTINUE
400 CONTINUE
4000 CONTINUE

```

```

c -----
c *** Back sweep
c -----

```

```

DO 5000 k = ku,kl,-1
  DO 5000 j = ju,jl,-1
    DO 5000 i = iu,il,-1
      DO 450 i1 = 1,no
        DO 450 k1 = 1,no
          t(i,j,k,i1) = t(i,j,k,i1)
&          - ae(i,j,k,i1,k1)*t(i+1,j,k,k1)
&          - an(i,j,k,i1,k1)*t(i,j+1,k,k1)
&          - at(i,j,k,i1,k1)*t(i,j,k+1,k1)
450 CONTINUE

```

```

C -----
C *** Store result in rh
C -----

      DO 460 j1 = 1,no
      rh(i,j,k,j1) = t(i,j,k,j1)
460      CONTINUE

5000 CONTINUE

C -----
C *** That's all, folks!
C -----

      RETURN
      END

C *****
C *
C * Subroutine: INDICES
C * Author: S. Babu
C * Version: 1.0
C * Date: December 1991
C *
C *****

      SUBROUTINE INDICES(il,iu,jl,ju,kl,ku,inp,indi,indj,indk)
C
C*****
C
C
C      indexing for vectorization
C
C*****
C
      INCLUDE 'parameter.inc'
      DIMENSION indi(idnp,ids),indj(idnp,ids),indk(idnp,ids),
1      inp(ids)

      nsurf = iu-il+ju-jl+ku-kl+1

```

```
DO isurf = 1,nsurf
  inp(isurf) =0
ENDDO
```

```
DO k = kl,ku
```

```
  DO j = j1,ju
```

```
    DO i = il,iu
```

```
      isurf = i+j+k-4
      inp(isurf) = inp(isurf) + 1
```

```
      indi(inp(isurf),isurf) = i
      indj(inp(isurf),isurf) = j
      indk(inp(isurf),isurf) = k
```

```
    ENDDO
```

```
  ENDDO
```

```
ENDDO
```

```
RETURN
```

```
END
```

```
C *****
C *
C * Subroutine: INV *
C * Author: K.H. Chen *
C * Revised by: S. Babu *
C * Version: 1.0 *
C * Date: December 1991 *
C *
C * 4X4 matrix inverse solver. LU decomposition is used. *
C *
C *****
```

```
SUBROUTINE INV(a)
```

```
INCLUDE 'parameter.inc'
```

```
DIMENSION a(4,4),ainv(4,4)
```

```
c -----
c *** Begin
c -----
```

```
rl11 = a(1,1)
rl21 = a(2,1)
rl31 = a(3,1)
rl41 = a(4,1)
```

```
u12 = a(1,2)/rl11
u13 = a(1,3)/rl11
u14 = a(1,4)/rl11
```

```
rl22 = a(2,2) - rl21*u12
rl32 = a(3,2) - rl31*u12
rl42 = a(4,2) - rl41*u12
```

```
u23 = (a(2,3) - rl21*u13)/rl22
u24 = (a(2,4) - rl21*u14)/rl22
```

```
rl33 = a(3,3) - rl31*u13 - rl32*u23
rl43 = a(4,3) - rl41*u13 - rl42*u23
```

```
u34 = (a(3,4) - rl31*u14 - rl32*u24)/rl33
```

```
rl44 = a(4,4) - rl41*u14 - rl42*u24 - rl43*u34
```

```
c -----
c *** Solve for new ainv
c -----
```

```
a(1,1) = 1.0d0/rl11
a(2,1) = (0.0d0 - rl21*a(1,1))/rl22
a(3,1) = (0.0d0 - rl31*a(1,1) - rl32*a(2,1))/rl33
a(4,1) = (0.0d0 - rl41*a(1,1) - rl42*a(2,1) - rl43*a(3,1))/rl44
```

```

a(1,2) = 0.0d0
a(2,2) = (1.0d0 - r121*a(1,2))/r122
a(3,2) = (0.0d0 - r131*a(1,2) - r132*a(2,2))/r133
a(4,2) = (0.0d0 - r141*a(1,2) - r142*a(2,2) - r143*a(3,2))/r144

```

```

a(1,3) = 0.0d0
a(2,3) = (0.0d0 - r121*a(1,3))/r122
a(3,3) = (1.0d0 - r131*a(1,3) - r132*a(2,3))/r133
a(4,3) = (0.0d0 - r141*a(1,3) - r142*a(2,3) - r143*a(3,3))/r144

```

```

a(1,4) = 0.0d0
a(2,4) = (0.0d0 - r121*a(1,4))/r122
a(3,4) = (0.0d0 - r131*a(1,4) - r132*a(2,4))/r133
a(4,4) = (1.0d0 - r141*a(1,4) - r142*a(2,4) - r143*a(3,4))/r144

```

```

c -----
c *** Back substitution
c -----

```

```

ainv(4,1) = a(4,1)
ainv(3,1) = a(3,1) - u34*ainv(4,1)
ainv(2,1) = a(2,1) - u23*ainv(3,1) - u24*ainv(4,1)
ainv(1,1) = a(1,1) - u12*ainv(2,1) - u13*ainv(3,1) - u14*ainv(4,1)

```

```

ainv(4,2) = a(4,2)
ainv(3,2) = a(3,2) - u34*ainv(4,2)
ainv(2,2) = a(2,2) - u23*ainv(3,2) - u24*ainv(4,2)
ainv(1,2) = a(1,2) - u12*ainv(2,2) - u13*ainv(3,2) - u14*ainv(4,2)

```

```

ainv(4,3) = a(4,3)
ainv(3,3) = a(3,3) - u34*ainv(4,3)
ainv(2,3) = a(2,3) - u23*ainv(3,3) - u24*ainv(4,3)
ainv(1,3) = a(1,3) - u12*ainv(2,3) - u13*ainv(3,3) - u14*ainv(4,3)

```

```

ainv(4,4) = a(4,4)
ainv(3,4) = a(3,4) - u34*ainv(4,4)
ainv(2,4) = a(2,4) - u23*ainv(3,4) - u24*ainv(4,4)
ainv(1,4) = a(1,4) - u12*ainv(2,4) - u13*ainv(3,4) - u14*ainv(4,4)

```

```

c -----

```

c *** Store result in a()

c -----

```
      DO 20 j = 1,4
        DO 10 i = 1,4
          a(i,j) = ainv(i,j)
10     CONTINUE
20     CONTINUE
```

c -----

c *** That's all, folks!

c -----

```
      RETURN
      END
```

APPENDIX C. PERFORMANCE ANALYSIS

USER PERFTRACE
ENVIRONMENT INFORMATION

ORIGINAL USER EXECUTION

OBSERVATIONS ABOUT YOUR PROGRAM

Whole Program

Observations:

This program achieved 95.5 million floating-point operations per second during its execution.

This program appears to be a partially vectorized code.

Routine CSIP3D

[Rank:1]

Observations:

This routine was responsible for 42.4% of the total program CPU time.

This is an important percentage of the whole program.
This routine achieved 129.9 million floating-point operations
per second during its execution.

This routine appears to be an efficient vectorized code.

Routine RHSMVIS

[Rank:2]

Observations:

This routine was responsible for 11.6% of the total program CPU time.
This routine achieved 119.8 million floating-point operations
per second during its execution.

This routine appears to be an efficient vectorized code.

**APPENDIX D. DIMENSIONAL ANALYSIS FOR THE SLOSHING
MOTION OF LIQUIDS IN CYLINDRICAL CONTAINERS
ROTATING ABOUT THE AXIS**

This appendix describes the process of identifying the nondimensional parameters [71] for the physical problem. The following variables are identified as the ones affecting the sloshing motion of liquids in partially filled cylindrical containers:

1. Radius of cylinder, r
2. Linear velocity of the cylinder wall, V_{ref}
3. Coefficient of viscosity of the liquid, μ
4. Surface tension, Γ
5. Density of liquid, ρ
6. Initial depth of liquid in the cylinder, h_0
7. Gravitational acceleration, g

As there are seven variables, $n = 7$; three dimensions, namely M , L , and T define these variables and hence $j = 3$. It is hoped that the problem can be described fully

by $4(n - j)$ 'pi' groups. The first step is to pick out the three variables that *do not* form a 'pi' group among themselves. These three can be identified as: r , V_{ref} and ρ .

The first 'pi' group is obtained by combining the viscosity μ with the above parameters.

$$\Pi_1 = V_{ref}^a \rho^b r^c \mu = M^0 L^0 T^0$$

Equating powers of M , L and T on both sides, it can be deduced that $a = -1$, $b = -1$, and $c = -1$. The inverse of such a combination is the Reynolds number, $Re = \frac{V_{ref} r}{\nu}$.

The other 'pi' groups follow by successively combining the other variables with the three independent variables.

$$\Pi_2 = V_{ref}^a \rho^b r^c g = M^0 L^0 T^0$$

which gives $\Pi_2 = \frac{V_{ref}^2}{gr}$.

$$\Pi_3 = V_{ref}^a \rho^b r^c \Gamma = M^0 L^0 T^0$$

which gives $\Pi_3 = \frac{\rho V_{ref}^2 r}{\gamma}$.

$$\Pi_4 = V_{ref}^a \rho^b r^c h = M^0 L^0 T^0$$

which gives $\Pi_4 = \frac{h}{r}$.

These parameters have been identified as conventional nondimensional numbers in fluid mechanics and listed in Table 4.1.

**APPENDIX E. DERIVATION OF THE STEADY STATE
FREE-SURFACE PROFILES FOR LIQUIDS INSIDE ROTATING
CYLINDERS: CASES WITH THE SURFACE TOUCHING THE TOP
WALL**

It can be shown [72] that for the case of rigid body rotation of fluid inside a cylindrical container, the pressure p is given by:

$$p = \frac{\rho\omega^2 r^2}{2} - \rho g + \text{constant}$$

and also, the free-surface height as a function of radius is given by:

$$h = \frac{\omega^2 r^2}{2g} + h_{min}$$

Let the radius at which the surface comes in contact with the cylinder top wall be ' r_t '. Hence, the following relation exists between the height of the cylinder, H and r_t :

$$H = \frac{\omega^2 r_t^2}{2g} + h_{min} \quad (\text{E.1})$$

Integration of the volume under the free-surface upto this radius gives:

$$V_1 = 2\pi \int_0^{r_t} r \left(\frac{\omega^2 r^2}{2g} + h_{min} \right) dr$$

$$V_1 = \frac{\pi\omega^2 r_i^4}{4g} + \pi h_{min} r_i^2$$

The volume contained in the cylinder beyond this radius is given by:

$$V_2 = \pi (R^2 - r_i^2) H$$

The sum of these two volume fractions must equal the initial volume inside the cylinder. Denoting the initial height of the fluid by h_0 , this volume is given by

$$V_0 = \pi R^2 h_0$$

Using the condition that $V_0 = V_1 + V_2$, and the equation E.1, the values of h_{min} and r_i and hence the free-surface profile given by 'h' can be found.