AN IMPLICIT PRESSURE-CORRECTION (SIMPLE) BASED FINITE-VOLUME/FRONT-TRACKING METHOD FOR COMPUTATIONS OF MULTIPHASE FLOWS IN COMPLEX GEOMETRIES

by

Satayef Kassabbashi

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Mechanical Engineering

Koç University

September, 2006

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Satayef Kassabbashi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

____

Assistant Prof. Metin Muradoğlu (Advisor)

____

Assistant Prof. Murat Sözer

____

Assistant Prof. Erdem Alaca

Date: ______________________

To my Family

ABSTRACT

Flows of gas-liquid or liquid-liquid mixed phases are familiar from many macroscopic systems. When discussing the multiphase flows in micro systems, there are likewise a myriad of questions arising in even simple configurations such as determination of the size of droplets of the dispersed phase, the speed at which the dispersed phase moves relative to the continuous phase and additional pressure drop in micro-channels due to presence of secondary phase. In this thesis, a finite-volume/front-tracking (FV/FT) method is developed for computations of multiphase flows in complex geometries as a toolbox to be used in designing and analysis of microfluidics systems. The method is based on the one field formulation of the flow equations and treating different phases as single fluid with variable material properties. The FV method is based on the SIMPLE algorithm on colocated body-fitted grids and is combined with the FT methodology. The interface is represented by connected Lagrangian marker points. A novel tracking algorithm is utilized to track the positions of front marker points in curvilinear grids and is found to be very robust and computationally efficient. The method is implemented to solve two-dimensional (plane or axisymmetric) dispersed multiphase flows. First, the particle tracking algorithm is tested and then the new FV/FT method is validated for the bouyancy-driven falling drop test case. Finally, the new method is used to compute the axisymmetric case of a gas phase displacing a Newtonean fluid.

Keywords: A finite-volume/front-tracking method, SIMPLE algorithm, dispersed multiphase flows, complex geometries, film deposition

ÖZET

Gaz-sıvı ve sıvı-sıvı faz akışları bir çok makroskopik sistemde görülmektedir. Mikro sistemlerdeki çok fazlı akışları incelerken, ayrık fazlardaki damla büyüklükleri, ayrık fazın sürekli faza göre olan rölatif hızı ve mikro kanallardaki basınç düşüşü en basit konfigürasyonda bile karşılaşılan problemlerdir. Bu çalışmada, çok fazlı akışkan sistemlerinin kompleks geometrilerde modellemesi için Sonlu Hacim/Arayüz izleme metodu geliştirilmiştir. Bu yöntemde, çok fazlı sistem, farklı malzeme özelliklerine sahip tek bir faz olarak modellenmiştir. Sonlu Hacim/Arayüz izleme metodu SIMPLE algoritması üstünde temellendirilmiş olup Arayüz izleme yöntemi ile birleştirilmiştir. Arayüz, birbirine bağlı Lagrange noktaları ile temsil edilmiştir. Bu noktaların yerlerini belirlemek için iyi çalışan bir tür izleme algoritması kullanılmıştır. Yeni geliştirilen bu metod, iki boyutlu ve aksisimetrik düzlemlerdeki çift fazlı akışkan problemlerini çözmek için implement edilmiştir. Bu çalışmada, öncelikle, tanecik izleme algoritması test edilmiş olup yeni Sonlu Hacim/Arayüz izleme metodu yerçekimi etkisinde düşen bir damla için test edilmiştir. Son olarak yeni geliştirilen Sonlu Hacim/Arayüz izleme metodu, aksisimetrik düzlemde gaz fazı ile Newtonian sıvının yer değiştirdiği durumun incelenmesinde kullanılmıştır.

Anahtar kelimeler: Sonlu Hacim/Arayüz izleme metodu, SIMPLE algoritması, ayrık çok fazlı akışkanlar, kompleks geometriler, film kaplaması

ACKNOWLEDGMENTS

First, I would like to thank my thesis advisor Dr. Metin Muradoğlu for his great supervision and continuous support. He shared his knowledge and experience with me, paid attention to all the steps of my thesis, and provided me his recommendations that guided me. I thank Dr. Murat Sözer and Dr. Erdem Alaca for their readings and comments for my thesis and for being in my thesis committee.

I thank Prof. M. Perić for providing us with the Computer Aided Fluid Flow Analysis (CAFFA) codes.

Also, I would like to thank all my friends at Koç University who made the two years an enjoyable period of time; my research group friends in particular Emrah Gölbaşi; my grad student friends in particular Emre Dikmen and all people who supported me all the time in the last two years.

All this would not be possible without the support of my family, I would like to thank them for their love, affection and support throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

Ca	Capillary number
CIP	Constrained Interpolation Profile
d	length scale
D	drop diameter
Δt	real-time step
$\Delta\rho$	density difference between the drop and the ambient fluids
Eo	Eötvös number
FD/FT	Finite Difference Front Tracking
FV/FT	Finite Volume Front Tracking
g_z	gravitational acceleration
γ	ratio of the dimensionless drop to the ambient fluid densities
h	film thickness
κ	dimensionless drop size
κ_c	twice of the mean curvature
λ	film length
μ_d	viscosity of the drop
ν	kinematic viscosity of the drop fluid
Oh_d	Ohnesorge number
p	pressure
r and z	radial and axial coordinates
R	radius of the tube
Re	Reynolds number

ρ_o	density of the ambient fluid
ρ_d	density of the drop
σ	surface tension coefficient
t	physical time
τ_{rr}, τ_{zz} and τ_{zr}	viscous stresses
U	velocity scale
v_r and v_z	velocity components in r and z coordinate directions
VOF	Volume-of-Fluid
$\mathbf{x}_f$	location of the front

Chapter 1

INTRODUCTION

Multiphase flows commonly occur in nature and in a multitude of other settings. They are not only of academic interest but are found in a wide range of engineering applications, continuing to pose a challenge to many research scientists and industrial practitioners alike. Although many important advances have been made in the past [3, 4, 5], the efforts to understand fundamental behavior and mechanisms of two-phase flow are necessarily a continuing process. Multiphase flows are encountered in both macro and microscopic systems. In macro systems, multiphase flows are encountered in oil recovery industry and in mixing/transport and deposition in energy flows. In oil recovery industry, displacement of oil drops trapped within the pores of rock formation by liquids or foams are just examples where modeling of multiphase flows in complex geometries becomes important. Flow of gas-liquid or liquid-liquid mixed phases are also encountered in microfluidic devices. In microfluidic devices it is of interest to determine the size of droplets of the dispersed phase or the speed at which the dispesed phase moves relative to the contiuous phase. Also, in microfluidic devices, it is of great importance to study the stability of the multiphase flow and the possibility of one phase to wet the boundaries and encapsulate the second fluid [3]. Both the physical modeling and the numerical computation of multiphase flows are associated with certain difficulties. These difficulties mainly arise from the existence of the moving interfaces separating the phases and immiscible fluids. In general, the geometry of the interfaces and the spatial distribution of the phases are not known as priori but are a part of the solution. The difficulties in modeling concern the physical transfer processes taking place across the interface such as momentum and phase change. The numerical difficulties arise from the fact that interface is moving and certain quantities are discontinuous across the interface, e.g., the density, viscosity and pressure [2].

First a short description of several terms relevant to multiphase flows will be given. A phase is a thermodynamic definition of matter, which can be either solid, liquid or gas. In a multiphase flow several phases flow together. These phases may consist of one chemical component, e.g., flow of water and water vapor (steam), or of several chemical components, e.g., flow of oil and water. Within the general multiphase terminology a phase is classified as continuous if it occupies continuously connected regions of space and is classified as disperse when it occupies disconnected regions of space. The disperse phase is formed by particles. In the sequel, particle can mean either a rigid (solid) particle or a fluid particle. Fluid particles formed by a gas phase are denoted as bubbles, while fluid particles formed by a liquid phase are called drops.

Any fluid motion originates from forces acting on fluid elements. A proper way to identify the dominant forces and those forces that may be neglected is provided by dimensionless groups, expressing the ratio between two forces. Four dimensionless groups, Reynolds number Re, Eötvös number Eo (interchangeably called the Bond number, Bo), the Ohnesorge number Oh_d, and the Capillary number Ca are important to mention. The Reynolds, Eötvös, Ohnesorge, and Capillary numbers represent the ratios between inertial and viscous forces, the buoyancy and surface tension forces, the viscous and inertial forces multiplied by the surface tension forces and the viscous and surface tension forces, respectively, and are defined as

$$\begin{aligned}
Re &= \frac{\rho_d d U}{\mu_d}, \\
E_o &= \frac{g_z \Delta\rho d^2}{\sigma}, \\
Oh_d &= \frac{\mu_d}{\sqrt{\rho_d d \sigma}}, \\
Ca &= \frac{\mu_d U}{\sigma},
\end{aligned} \tag{1.1}$$

where ρ_d and μ_d denote the density and viscosity of the drop fluid; ρ_o and μ_o denote the density and viscosity of the ambient fluid; $\Delta\rho = \rho_d - \rho_o$ and σ denote the density difference between the drop and the ambient fluids, and the surface tension coefficient; g_z, U and d denote the gravitational acceleration, the velocity and length scales, respectively. The parameters that govern the motion of a freely suspended liquid drop through a capillary at low Reynolds number are the size of the drop relative to the diameter of the tube, the ratio of the viscosity of the drop fluid to the viscosity of the ambient fluid, and the capillary number, as mentioned above. These three parameters also affect the relative mobilities of fluid phases in flow through porous media, but the capillary number holds special significance because it determines whether trapped drops can be mobilized and displaced through constrictions.

Especially, the motion of immiscible liquid drops and bubbles suspended in low Reynolds number flow through a capillary tube is a prototype problem that has proven useful in analyzing multiphase fluid systems in a variety of flows [3, 4, 5]. Studies of the motion of drops and bubbles in capillaries provide insight into pore-scale hydrodynamics that cannot be scrutinized in practice.

The presence of deforming phases makes the multiphase flow computations a challenging problem and strong interactions between the phases and complex boundaries add further complexity to the problem. Therefore, the progress was rather slow and the computations of multiphase flows have been usually restricted to simple geometries and boundary conditions. Since nearly all multiphase flows of practical interest involve complex geometries and boundary conditions, it is of obvious interest to extend the modeling and computational techniques to treat multiphase flows in complex geometries.

The interaction of droplets with the geometry in which they are constrained to traverse motion has received some attention from numerical (Tsai and Miksis [5], Manga [6], Udaykumar et al. [7]) and experimental viewpoints (Olbricht and Leal [8], Hemmat and Borhan [9]). Since the major focus has been on the behavior of multiphase flow through porous media with applications in oil recovery, such flows have been restricted to the creeping flow regime. The boundary integral technique was applied to simulate such a problem by Tsai and Miksis [5]. In a recent work, Manga [6] applied the same technique to study the behavior of drops in geometries such as a driven cavity and bifurcating channel, again restricted to Stokes flow. Udaykumar et al. [7] studied the deformation of droplets in viscosity and inertia dominated flows through a constricted tube and reproduced the results obtained by Tsai and Miksis [5].

A number of experimental studies have examined the motion of liquid droplets through sinusoidally constricted capillary tubes, however the most relevant ones have been performed by Olbricht and Leal [8] and Hemmat and Borhan [9]. Olbricht and Leal [8] considered the pressure driven motion of immiscible neutrally-buoyant drops in pressure driven flow through horizontal capillaries whose diameter changes periodically with the axial position. They measured the average drop speed and correlated those values with the observed time-dependant drop shapes. Their results were mainly qualitative and they demonstrated the effect of capillary geometry on the shape and mobility of drops. Hemmat and Borhan [9] reported the buoyancy driven motion of viscous drops and gas bubbles through a vertical capillary with periodic variations in order to examine the effect of capillary geometry on drop deformation and breakup.

Simulation of multiphase flows is notoriously difficult mainly due to the presence of deforming phase boundaries. A variety of numerical methods have been developed and successfully applied to a wide range of multifluid and multiphase flow problems [10, 11, 12, 13, 14]. In spite of this success, significant progress is still needed especially for accurate computations of multiphase flows involving strong interactions with complex solid boundaries.

The most popular approaches to compute the multiphase flows are classified into four categories: The first category is the front capturing method such as the Volume-Of-Fluid (VOF) [11, 15] and the level-set [10, 12, 16] methods. In these methods, the front is captured indirectly through the volume-fraction distribution (VOF) or the zero-level-set of the distance function. The constrained interpolation profile (CIP) method of Yabe [14] also belongs to this category. Traditionally, the main difficulty in using these methods has been the maintenance of a sharp boundary between the different fluids and the computation of the surface tension. The second class of methods, and the one that offers the potentially highest accuracy, uses separate, boundary fitted grids for each phase. The steady rise of buoyant, deformable, axisymmetric bubbles was simulated by Ryskin and Leal [17] using this method. This method is best suited for relatively simple geometries, and applications to complex fully three-dimensional problems with unsteady deforming phase boundaries are very rare. The third class is the Lagrangian methods with moving grids [18]. The fourth approach used in the present study is the front-tracking method [13, 19]. The front-tracking method developed by Unverdi and Tryggvason [19] is based on the one-field formulation of the Navier-Stokes equations and treating different phases as a single fluid with variable material properties. In this approach, a stationary Eulerian grid is used for the fluid flow and the interface is tracked explicitly by a separate Lagrangian grid. The immersed boundary method developed by Peskin [20] is used to smoothly discretize the jumps in material properties and to treat the effects of surface tension. The front-tracking method combined with a finite-difference flow solver has been successfully applied to a wide range of multiphase flow problems but almost all in relatively simple geometries [13] except for Udaykumar et al. [7]. Udaykumar et al. [7] used a mixed Eulerian-Lagrangian algorithm to compute flows in arbitrary geometries. The front tracking method has been reviewed recently by Tryggvason et al. [13].

It is of great importance to be able to accurately model strong interactions between bubbles/drops and the curved solid boundary in many engineering and scientific applications such as microfluidic systems [3], pore-scale multi-phase flow processes [8, 21] and biological systems [4, 22]. The front-tracking method has many advantages such as its conceptual simplicity and lack of numerical diffusion. However, its main disadvantage is probably the difficulty to maintain the communication between the Lagrangian marker points and Eulerian body-fitted curvilinear or unstructured grids. In the present study, a finite-volume/front tracking (FV/FT) method is developed to compute dispersed multiphase flows in complex geometries. FV/FT method with the FV solver being based on the concept of dual time-stepping was implemented earlier for the computation of two dimensional (plane or axisymmetric) dispersed flows in complex geometries [1, 23]. Two-dimensional plane results have been published earlier by Muradoglu and Gokaltun [23], and the axisymmetric flow cases are presented by Muradoglu and Kayaalp [1]. The FV solver utilized in FV/FT method by Muradoglu and Kayaalp is a density based solver that uses a preconditioning method. Muradoglu and Kayaalp implemented the FV/FT method to solve the buoyancy-driven falling drops in a straight channel and the motion of buoyancy-driven drops in capillary constricted tube. The density based FV solver is found to be too sensitive to relaxation parameters as being based on the preconditioning method and thus not very robust.

In the present work, the density based FV solver is replaced by a SIMPLE type FV solver and the front-tracking FT methodology is extended to colocated curvilinear grids and is combined with SIMPLE method [24] to facilitate accurate and efficient modeling of strong interactions between the phases and complex solid boundary. Many solution methods for incompressible flows are of the implicit pressure-correction type. They use a pressure-correction equation to enforce mass conservation at each time step. Being an implicit scheme, the SIMPLE algorithm is prefered for slow transient flows because it has less stringent time step restrictions than explicit scheme. The SIMPLE solver is less sensitive to relaxation parameters which makes the algorithm more robust and therefore easier to use by non-experts. For this reason, SIMPLE algorithm is prefered and utilized in nearly all commercial CFD packages including FLUENT, StarCD and CFX. The FV algorithm is essentially based on the cell-centered finite-volume scheme implemented in *caffa* code [24]. The code has been downloaded from the internet and modified to include the computation of the surface tension forces resulting from the curvature of the interface separating the immiscible fluids. Then, the grid generator associated with the *caffa* code is modified to generate complex solid boundaries and to maintain the communication between the curvilinear and Lagrangian grids. Being associated with complex grid generator, the new FV/FT method is capable of computing flows with very large gradients at the boundaries such as the deposition of a fluid on the wall of a tube problem studied theoretically by Taylor [25] and Bretherton [26] and experimentally by Aussillous and Qu*éré* [27].

The thesis is organized into three main parts. In Chapter 2, the governing equations, discretization schemes and the implicit pressure correction algorithm are briefly reviewed. Then the finite-volume/front-tracking algorithm is described. In Chapter 3, the particle tracking algorithm is tested in a simple setting of a rigid body rotation of fluid in a circular channel and then the new FV/FT method is applied to compute the motion of the drops falling due to gravity in a straight channel studied earlier numerically by Muradoglu and Kayaalp [1] and by Han and Tryggvason [2]. It is found that the present results are in a reasonably good agreement with the results obtained by early researchers. Then, the present method is adopted to study the axisymmetric case of a gas phase steadily displacing a Newtonian liquid and the results are compared with those obtained by early investigators [27, 26, 28]. Finally, some conclusions are drawn in Chapter 4.

Chapter 2

NUMERICAL METHOD

In this chapter, an implicit finite volume scheme that uses the pressure-correction method on two-dimensional Cartesian grid is presented. Solution in complicated geometries is described by Ferziger and Peric [24]. First the mathematical formulation is explained in details. Then, the implicit pressure correction method is described. Finally the finite-volume/front-tracking algorithm is presented.

2.1 Mathematical Formulation

The FV solver is essentially based on the cell-centered scheme implemented in *caffa* code [24]. The code is downloaded from the internet and is adopted to solve the flow equations including the effects of the surface tension forces. The governing Navier-Stokes equations in integral form read:

$$\int_S \rho \boldsymbol{v}.\boldsymbol{n} dS = 0, \tag{2.1}$$

$$\frac{\partial}{\partial t}\int_\Omega \rho u_i d\Omega + \int_S \rho u_i \boldsymbol{v}.\boldsymbol{n} dS = \int_S \tau_{ij}\boldsymbol{i}_j.\boldsymbol{n} dS - \int_S p\boldsymbol{i}_i.\boldsymbol{n} dS + \int_\Omega f_{b_i} d\Omega \tag{2.2}$$

where S and Ω refer to the surface integral over the CV face and the volume integral over the CV, respectively. All terms appearing in the above equations are

ρ density

$\boldsymbol{n}$ unit normal vector

$\boldsymbol{v}$ velocity field

u_i the i^{th}component of the velocity field

τ_{ij} viscous stress tensor

i_j j^{th} component of unit vector

p pressure

f_b body force

Equation (2.1) is the integral form of the mass conservation (continuity) equation. Terms

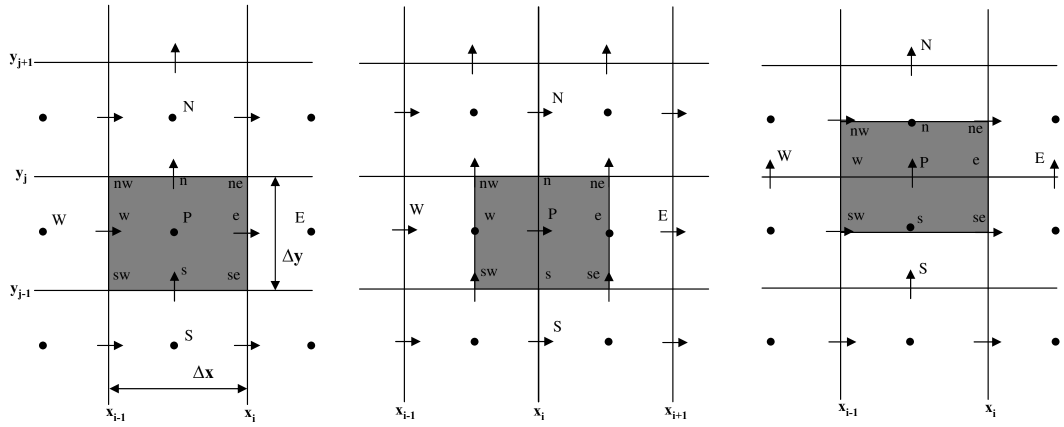


Figure 2.1: Control volumes for a staggered grid: for mass conservation and scalar quantities (left), for x-momentum u-CV (center) and for y-momentum v-CV (right).

on the left side of the integral form of the momentum equation Eq. (2.2) represent the unsteady and convective fluxes, respectively. On the right hand side, the momentum flux is split into a viscous contribution $\tau_{ij}\mathbf{i_j}$ and a pressure contribution $p\mathbf{i_i}$. The last term f_{b_i} corresponds to the body force including gravitational and surface tension forces. Typical staggered and colocated control volumes are shown in Fig. (2.1) and Fig. (2.2), respectively. In staggered grid arrangement, the control volumes for u_x and u_y are displaced with respect to the control volume for the continuity equation as shown in Fig. (2.1). Implicit solution of the momentum equation discretized with colocated arrangement is the same as for staggered arrangement except that CVs are the same for all variables. Cell faces e and w for u_x and n and s for u_y lie midway between the nodes. For convenience, u is sometimes used instead of u_x and v instead of u_y. A second order implicit three time level scheme is used for the integration in time. This leads to the following approximation of the unsteady term:

$$\left[\frac{\partial}{\partial t}\int_{\Omega}\rho u_i d\Omega\right]_P \approx \frac{\rho\Delta\Omega}{2\Delta t}(3u_i^{n+1} - 4u_i^n + u_i^{n-1}) = A_P^t u_{i,P}^{n+1} - Q_{u_i}^t, \tag{2.3}$$

where

$$A_P^t = \frac{3\rho\Delta\Omega}{2\Delta t} \quad \text{and} \quad Q_{u_i}^t = \frac{\rho\Delta\Omega}{2\Delta t}(4u_i^n - u_i^{n-1}). \tag{2.4}$$

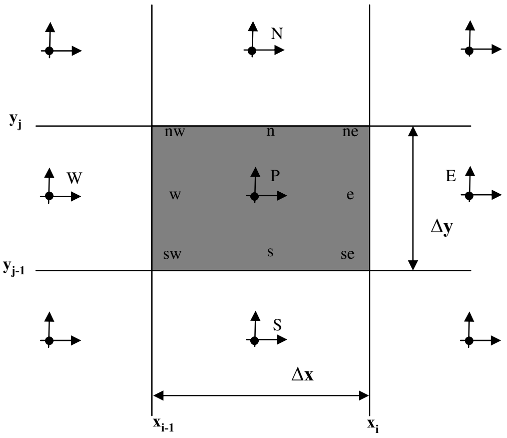


Figure 2.2: Control volume in colocated grid and notation used.

The superscript $n+1$ is dropped since all terms are evaluated at t_{n+1} unless stated otherwise. Because the scheme is implicit, the equations require iterative solution. Outer iterations are to be distinguished from inner iterations used to solve linear equations such as the pressure correction equation. For the inner iterations, one of the basic solvers for linear equation systems is used and concentration shall be on the outer iterations. The surface integrals are splitted into four CV face integrals; east, west, north and south faces are denoted by e, w, n and s, respectively. Approximation of the convective and diffusive fluxes and the source terms are considered on CV face e; the other faces are treated in the same way, and the results can be obtained by index substitution. A second order central difference approximation is adopted to approximate the spatial derivatives appearing in convective and diffusive fluxes. Fluxes are approximated by assuming that the value of a quantity at a CV face center represents the mean value over the face (mid-point rule approximation). In the m^{th} outer iteration all nonlinear terms are approximated by a product of an *old* (from the preceding outer iteration) and a *new* value. Thus, in discretizing the momentum equations, the mass flux through each CV face is evaluated using the existing velocity field and is assumed known:

$$\dot{m}_e^m = \int_S \rho \boldsymbol{v}.\boldsymbol{n} dS \approx (\rho u)_e^{m-1} S_e. \tag{2.5}$$

For convenience, all varaiables in the remainder of this section belong to the m^{th} outer iteration unless specified otherwise. The mass fluxes (Eq. (2.5)) satisfy the continuity equation on the scalar CV (Fig. (2.1)). Mass fluxes at the faces of the momentum CVs are obtained by linear interpolation. Since the east and west faces of a u-CV are half way between scalar CV faces, the mass fluxes are calculated as:

$$\dot{m}_e^u = \frac{1}{2}(\dot{m}_P + \dot{m}_E)^u; \quad \dot{m}_w^u = \frac{1}{2}(\dot{m}_W + \dot{m}_P)^u. \tag{2.6}$$

The mass fluxes through the north and south faces of the u-CV are approximated as half the sum of the two scalar CV face mass fluxes:

$$\dot{m}_n^u = \frac{1}{2}(\dot{m}_{ne} + \dot{m}_{nw})^u; \quad \dot{m}_s^u = \frac{1}{2}(\dot{m}_{se} + \dot{m}_{sw})^u. \tag{2.7}$$

The superscript u denotes that the indices refer to the u-CV (Fig. (2.1)). The sum of four mass fluxes for the u-CV is half of mass fluxes into two adjacent scalar CVs and therefore satisfy the continuity equation for the double scalar CV so the mass fluxes through the u-CV faces conserve mass. This result also holds for v-momentum CVs ensuring that the mass fluxes through the momentum CVs satisfy the continuity equation. The convective flux of u-momentum through e face of a u-CV is then:

$$F_{i,e}^c = \int_S \rho u_i \boldsymbol{v}.\boldsymbol{n} dS \approx \dot{m}_e u_{i,e}. \tag{2.8}$$

The CV face value of u_i used in the above expression can be found using simple linear interpolation *central difference scheme* (CDS), but some iterative solvers fail to converge when applied to algebraic equation systems derived from central difference approximations of convective fluxes because the matrices may not be diagonally dominant. Thus, these equations are solved using *deferred correction* approach in which the flux is expressed as

$$F_{i,e}^c = \dot{m}_e u_{i,e}^{UDS} + \dot{m}_e (u_{i,e}^{CDS} - u_{i,e}^{UDS})^{m-1}, \tag{2.9}$$

where superscripts CDS and UDS denote approximations by central and upwind differences, respectively. The outward unit normal vector at CV face e is $\boldsymbol{i}$, and thus the diffusive fluxes are calculated as:

$$F_{i,e}^d = \int_{S_e} \tau_{ix} dS \approx (\tau_{ix})_e S_e, \tag{2.10}$$

where $S_e = y_j - y_{j-1} = \Delta y$ for the u-CV and $S_e = \frac{1}{2}(y_{j+1} - y_{j-1})$ for the v-CV. The stresses at CV faces are calculated by approximating the derivatives using central difference scheme as:

$$(\tau_{xx})_e = 2(\mu\frac{\partial u}{\partial x})_e \approx 2\mu\frac{u_E - u_P}{x_E - x_P}, \tag{2.11}$$

$$(\tau_{yx})_e = \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})_e \approx \mu\frac{v_E - v_P}{x_E - x_P} + \mu\frac{u_{ne} - v_{se}}{y_{ne} - y_{se}}, \tag{2.12}$$

where τ_{xx} is evaluated at the e face of the u-CV, and τ_{yx} at the e face of the v-CV.

The pressure terms are approximated by:

$$Q_u^p = -\int_S p\boldsymbol{i.n}dS \approx -(p_e S_e - p_w S_w)^{m-1} \tag{2.13}$$

for the u-equation and

$$Q_v^p = -\int_S p\boldsymbol{j.n}dS \approx -(p_n S_n - p_s S_s)^{m-1} \tag{2.14}$$

for the v-equation.

Body forces are approximated by:

$$Q_{u_i}^b \approx (f_b)^{m-1}\,\Delta\Omega \tag{2.15}$$

where $\Delta\Omega = (x_e - x_w)(y_n - y_s) = \frac{1}{2}(x_{i+1} - x_{i-1})(y_j - y_{j-1})$ for the u-CV and $\Delta\Omega = \frac{1}{2}(x_i - x_{i-1})(y_{j+1} - y_{j-1})$ for the v-CV. The aproximation to the complete u_i-momentum equation is:

$$A_P^t u_{i,P} + F_i^c = F_i^d + Q_i^p + Q_i^b + Q_i^t, \tag{2.16}$$

where

$$F^c = F_e^c + F_w^c + F_n^c + F_s^c \quad \text{and} \quad F^d = F_e^d + F_w^d + F_n^d + F_s^d \tag{2.17}$$

Since ρ and μ are constant, part of the diffusive flux term cancels out by the virtue of the continuity equation. The equations are simplified by deleting those terms prior to discretization. In the u-equation, τ_{xx} term on e and w faces are reduced by half, and in τ_{yx} term at n and s faces $\partial v/\partial x$ contribution is removed. Even when ρ and μ are not constant, the sum of these terms contributes in only a minor way to F^d. This is why an *explicit diffusive source* term for u:

$$Q_u^d = \left[\mu_e S_e\frac{u_E - u_P}{x_E - x_P} - \mu_w S_w\frac{u_P - u_W}{x_P - x_W} + \mu_n S_n\frac{v_{ne} - v_{nw}}{x_{ne} - x_{nw}} - \mu_s S_s\frac{v_{se} - v_{sw}}{x_{se} - x_{sw}}\right]^{m-1} \tag{2.18}$$

is usually calculated from the previous outer iteration $m-1$ and treated explicitly. Only $F^d - Q_u^d$ is treated implicitly. A consequence of this approximation is that, on a colocated grid, the matrix implied by Eq. (2.16) is identical for all velocity components. When the approximation for all the fluxes and source terms are substituted into Eq.(2.16), the following form of algebraic equation is obtained:

$$A_P^u u_P + \sum_l A_l^u u_l = Q_P^u, \quad l = E, W, N, S. \tag{2.19}$$

The equation for v has the same form. For the CDS approximations applied above, the coefficients of the u equation are:

$$\begin{aligned} A_E^u &= min(\dot{m}_e^u, 0) - \frac{\mu_e S_e}{x_E - x_P}, \\ A_N^u &= min(\dot{m}_n^u, 0) - \frac{\mu_n S_n}{y_N - y_P}, \\ A_W^u &= min(\dot{m}_w^u, 0) - \frac{\mu_w S_w}{x_P - x_W}, \\ A_S^u &= min(\dot{m}_s^u, 0) - \frac{\mu_s S_s}{y_P - y_S}, \\ A_P^u &= A_P^t - \sum_l A_l^u, \quad l = E, W, N, S. \end{aligned} \tag{2.20}$$

The source term Q_P^u contains not only the pressure and body force terms but also the portion of convective and diffusive fluxes resulting from deferred correction and the contribution to the unsteady term:

$$Q_P^u = Q_u^p + Q_u^b + Q_u^c + Q_u^d + Q_u^t, \tag{2.21}$$

where

$$Q_u^c = \left[(F_u^c)^{UDS} - (F_u^c)^{CDS}\right]^{m-1}. \tag{2.22}$$

The *convective source* is calculated using the velocities from the previous outer iteration $m-1$. The coefficients for the v-equation are obtained in the same way and have the same form. The linearized momentum equations are solved with sequential solution method [24] using the *old* mass fluxes and the pressure from the previous outer iteration. This produces new velocities u^* and v^* which do not necessarily satisfy the continuity equation so:

$$\dot{m}_e^* + \dot{m}_w^* + \dot{m}_n^* + \dot{m}_s^* = \Delta\dot{m}_P^*, \tag{2.23}$$

where the mass fluxes are calculated according to Eq. (2.5) using u^* and v^*. The velocity components u^* and v^* calculated from the momentum equatios can be expressed as follows (by dividing Eq. (2.19) by A_P; noting that index e on a mass CV represents index P on a u-CV):

$$u_e^* = \tilde{u}_e^* - \frac{S_e}{A_P^u}(p_E - p_P)^{m-1}, \tag{2.24}$$

where $\tilde{u}_e^*$ is shorthand notation for

$$\tilde{u}_e^* = \frac{Q_P^u - Q_u^p - \sum_l A_l^u u_l^*}{A_P}. \tag{2.25}$$

Analogously, v_n^* is expressed as:

$$v_n^* = \tilde{v}_n^* - \frac{S_n}{A_P^v}(p_N - p_P)^{m-1}, \tag{2.26}$$

The velocities u^* and v^* are corrected to enforce mass conservation by correcting the pressure. This is done as outlined in the following section. The corrected velocities $u^m = u^* + u^{'}$ and $v^m = v^* + v^{'}$,where $u^{'}$ and $v^{'}$ refer to the velocity corrections, are enforced to satisfy linearized momentum equations by correcting the pressure. Therefore

$$u_e^m = \tilde{u}_e^m - \frac{S_e}{A_P^u}(p_E - p_P)^m, \tag{2.27}$$

and

$$v_n^m = \tilde{v}_n^m - \frac{S_n}{A_P^v}(p_N - p_P)^m, \tag{2.28}$$

where $p^m = p^{m-1} + p^{'}$ is the new pressure and $p^{'}$ refers to the pressure correction. The relation between the velocity and pressure corrections is obtained by substracting Eq. (2.24) from Eq. (2.27):

$$u_e^{'} = \tilde{u}_e^{'} - \frac{S_e}{A_P^u}(p_E^{'} - p_P^{'}), \tag{2.29}$$

where

$$\tilde{u}_e^{'} = \tilde{u}_e^m - \tilde{u}_e^* = -\frac{\sum_l A_l^u u_l^{'}}{A_P}. \tag{2.30}$$

Anologously,

$$v_n^{'} = \tilde{v}_n^{'} - \frac{S_n}{A_P^v}(p_N^{'} - p_P^{'}). \tag{2.31}$$

The velocities u^m and v^m are substituted into the expressions for mass fluxes (Eq.(2.5)) to enforce to satisfy the continuity equation; and using Eq.(2.23):

$$(\rho S u^{'})_e - (\rho S u^{'})_w + (\rho S v^{'})_n - (\rho S v^{'})_s + \Delta \dot{m}_P^* = 0. \tag{2.32}$$

Finally, the above expressions for $u^{'}$ and $v^{'}$ are substituted into the continuity equation leading to the pressure correction equation:

$$A_P^p p_P^{'} + \sum_l A_l^p p_l^{'} = -\Delta \dot{m}_P^* - \Delta \dot{m}_P^{'}, \tag{2.33}$$

where the coefficients are:

$$\begin{aligned} A_E^p &= -\left(\frac{\rho S^2}{A_P^u}\right)_e, & A_W^p &= -\left(\frac{\rho S^2}{A_P^u}\right)_w, \\ A_S^p &= -\left(\frac{\rho S^2}{A_P^v}\right)_n, & A_S^p &= -\left(\frac{\rho S^2}{A_P^v}\right)_s, \\ A_P^p &= -\sum_l A_l^p, & l &= E, W, N, S. \end{aligned} \tag{2.34}$$

The term $\Delta \dot{m}_P^{'}$ is analogous to $\Delta \dot{m}_P^*$, with $\tilde{u}^{'}$ and $\tilde{v}^{'}$ replacing u^* and v^*. Since the velocity corrections are not known prior to the solution of the pressure-correction equation, this term is neglected resulting in the SIMPLE algorithm. Other alternatives are the SIMPLER and the PISO [24]. After the pressure-correction equation is solved, the velocities and pressure are corrected. The corrected velocities satisfy the continuity equation to the accuracy with which the pressure-correction equation is solved. However, they do not satisfy the non-linear momentum equation, so another outer iteration is needed. When both the continuity and momentum equations are satisfied to the desired accuracy, one can proceed to the next level.

2.2 Implicit Pressure-Correction Method

If an implicit method is used to advance the momentum equations in time, the discretized equations for the velocities at the new time step are non-linear. If the pressure gradient term is not included in the source term, these may be written:

$$A_P^{u_i} u_{i,P}^{n+1} + \sum_l A_l^{u_i} u_{i,l}^{n+1} = Q_{u_i}^{n+1} - \left(\frac{\delta p^{n+1}}{\delta x_i}\right)_P. \tag{2.35}$$

The source term contains all the terms that may be explicitly computed in terms of u_i^n as well as any body force or other linearized terms that may depend on u_i^{n+1} or other variables at the new time level - hence the superscript $n+1$. Due to the non-linearity and coupling of the uderlying differential equations, Eq. (2.35) cannot be solved directly as the coefficients

A and, possibly, the source term, depend on the unknown solution u_i^{n+1}. The iterations within one time step, in which the coefficient and source matrices are updated, are called *outer iterations* to distinguish them from the *inner iterations* performed on linear system with fixed coefficients. On each outer iteration, the following system of linear equations is solved:

$$A_P^{u_i} u_{i,P}^{m*} + \sum_l A_l^{u_i} u_{i,l}^{m*} = Q_{u_i}^{m-1} - \left(\frac{\delta p^{m-1}}{\delta x_i} \right)_P . \tag{2.36}$$

At the beginning of each outer iteration, the terms on the right hand side of Eq. (2.36) are evaluated using the variables at the preceeding outer iteration. Since the pressure used in these iterations was obtained from the previous outer iteration or time step, the velocities computed from Eq. (2.36) do not normally satisfy the continuity equation. To enforce the continuity condition, the velocities need to be corrected; this requires modification of the pressure field.

The velocity at node P, obtained by solving the linearized momentum equations, can be formally expressed as

$$u_{i,P}^{m*} = \frac{Q_{u_i}^{m-1} - \sum_l A_l^{u_i} u_{i,l}^{m*}}{A_P^{u_i}} - \frac{1}{A_P^{u_i}} \left(\frac{\delta p^{m-1}}{\delta x_i} \right)_P \tag{2.37}$$

where $u_{i,P}^{m*}$ is a predicted value of the velocity for iteration m. The corrected final values should satisfy the continuity equation. For convenience, the first term on the right hand side of the above equations is called $\tilde{u}_{i,P}^{m*}$ defined as

$$u_{i,P}^{m*} = \tilde{u}_{i,P}^{m*} - \frac{1}{A_P^{u_i}} \left(\frac{\delta p^{m-1}}{\delta x_i} \right)_P . \tag{2.38}$$

The velocities are corrected so that they satisfy the continuity equation by correcting the pressure field. The corrected velocities and pressure are linked by

$$u_{i,P}^{m} = \tilde{u}_{i,P}^{m*} - \frac{1}{A_P^{u_i}} \left(\frac{\delta p^{m}}{\delta x_i} \right)_P . \tag{2.39}$$

Continuity is now enforced by inserting the above expression for $u_{i,P}^m$ into the continuity equation to yield a discrete Poisson equation for the pressure

$$\frac{\delta}{\delta x_i} \left[\frac{\rho}{A_P^{u_i}} \left(\frac{\delta p^m}{\delta x_i} \right) \right]_P = \left[\frac{\delta \left(\rho \tilde{u}_i^{m*} \right)}{\delta x_i} \right]_P . \tag{2.40}$$

After solving the Poisson equation for the pressure, the final velocity field at the new iteration, u_i^m, is calculated from Eq. (2.39). At this point the velocity field satisfies the continuity equation, but the velocity and pressure fields do not satisfy the momentum equations. Thus, another outer iteration in needed and the process continues until a velocity field which satisfies both the momentum and continuity equations is obtained. In SIMPLE algorithm, the most common implicit pressure-correction method, a small correction is added to the velocities computed from the linearized momentum equations and the pressure p^{m-1}, i.e.,

$$u_i^m = u_i^{m*} + u^{'} \quad \text{and} \quad \mathrm{p^m = p^{m-1} + p^{'}}. \tag{2.41}$$

These corrected values are substituted into the momentum equation Eq. (2.36), and the relation between the velocity and pressure corrections is obtained as

$$u_{i,P}^{'} = \tilde{u}_{i,P}^{'} - \frac{1}{A_P^{u_i}} \left(\frac{\delta p^{'}}{\delta x_i} \right)_P , \tag{2.42}$$

where $\tilde{u}_{i,P}^{'}$ is defined as

$$\tilde{u}_{i,P}^{'} = - \frac{\sum_l A_l^{u_i} u_{i,l}^{'}}{A_P^{u_i}}. \tag{2.43}$$

Application of the discretized continuity equation to corrected velocities and use of Eq. (2.42) produces the following pressure-correction equation

$$\frac{\delta}{\delta x_i} \left[\frac{\rho}{A_P^{u_i}} \left(\frac{\delta p^{'}}{\delta x_i} \right) \right]_P = \left[\frac{\delta \left(\rho u_i^{m*} \right)}{\delta x_i} \right]_P + \left[\frac{\delta \left(\rho \tilde{u}_i^{'} \right)}{\delta x_i} \right]_P . \tag{2.44}$$

Once the pressure correction has been solved for, the velocities are updated using Eq. (2.41) and Eq. (2.42). The above procedure is repeated using the new values of u_i^m and p^m as improved estimates for u_i^{n+1} and p^{n+1} until all corrections are negligibly small, thus one can advance to the next time step [24].

The solution algorithm for the SIMPLE method can be summarized as follows:

1. Start calculation of the fields at the new time t_{n+1} using the latest solution u_i^n and p^n as starting estimates for u_i^{n+1} and p^{n+1}.

2. Assemble and solve the linearized algebraic equation systems (Eq. (2.36)) for the velocity components (momentum equations) to obtain u_i^{m*}.

3. Assemble and solve the pressure-correction equation (Eq. (2.44)) to obtain $p^{'}$.

4. Correct the velocities and pressure to obtain the velocity field u_i^m, which satisfies the continuity equation, and the new pressure p^m.

5. Return to step 2 and repeat, using u_i^m and p^m as improved estimates for u_i^{n+1} and p^{n+1}, until all corrections are negligibly small.

6. Advance to the next time step.

2.3 Tracking Algorithm

In the present work, the tracking algorithm developed by Muradoglu and Kayaalp [1] is utilized and combined with the FV solver. In this tracking algorithm, the interface between different phases are represented by a Lagrangian grid with connected marker points as shown in Fig. (2.3). The marker points can be considered as fluid particles moving with local flow velocity. In order to maintain communication between the Lagrangian and fixed curvilinear grids, it is necessary to determine the locations of the marker points in the curvilinear grid at every physical time step. The grid generator associated with the *caffa* code [24] is modified to maintain the comunication three different grids (curvilinear grid, auxilary uniform cartesian grid and lagrangian grid) Fig. (2.3). The tracking algorithm utilizes an auxiliary uniform Cartesian grid as sketched in Fig. (2.3). The overall algorithm can be summarized as follows: At the beginning of each simulation, a uniform Cartesian grid is generated such that it covers the entire computational domain. The cell size of the uniform grid is typically taken as the half of the size of the smallest curvilinear grid cell. It is then found which uniform Cartesian grid nodes reside in each curvilinear grid cell and this information is stored in an array.

Referring to the sketch in Fig. (2.4), for example, the nodal point Q is found to be in the curvilinear grid cell $ABCD$ by performing the vector operations $\overrightarrow{k} \cdot (|\overrightarrow{AB}| \times |\overrightarrow{AQ}|) \geq 0$, $\overrightarrow{k} \cdot (|\overrightarrow{BC}| \times |\overrightarrow{BQ}|) \geq 0$, $\overrightarrow{k} \cdot (|\overrightarrow{CD}| \times |\overrightarrow{CQ}|) \geq 0$, and $\overrightarrow{k} \cdot (|\overrightarrow{DA}| \times |\overrightarrow{DQ}|) \geq 0$, where $\overrightarrow{k}$ is the outward unit vector perpendicular to the cell $ABCD$. The same procedure is used for all other nodal points enclosed by the rectangle consisting of the uniform grid cells that enclose $ABCD$, i.e., the uniform grid cells in the rectangle shown by thick solid borderline in the sketch, and the entire process is repeated for all other curvilinear grid cells. It is emphasized that all these computations are done only once at the beginning of each

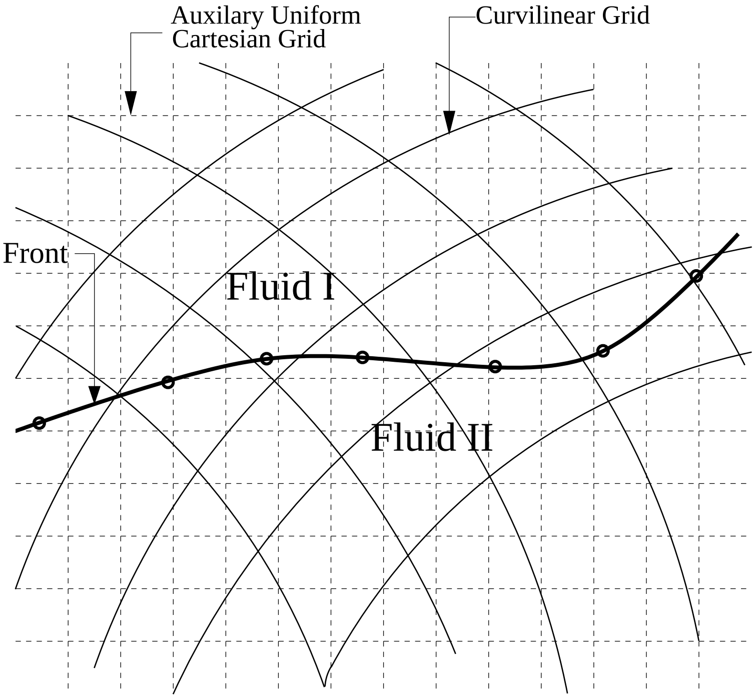


Figure 2.3: Three types of grids used in the computations. The governing equations are solved on a fixed Eulerian curvilinear grid and the interface between different phases is represented by a Lagrangian grid consisting of connected marker points. An auxiliary uniform Cartesian grid is used to maintain communication between the curvilinear and Lagrangian grids [1].

simulation as a preprocessing. Then, in each physical time step, it is first determined where the front points reside in the uniform grid. Referring to the sketch in Fig. (2.5), for instance, it is first found that the front point P is in (I, J) cell of the uniform grid and then it is determined that the nodes of (I, J) cell reside in the curvilinear grid cells (i, j), $(i, j-1)$ and $(i-1, j-1)$ as shown by the dashed line in the sketch. As a result, we conclude that the front point P resides in the region consisting of the curvilinear grid cells $(i-1 : i, j-1 : j)$ and this region is expanded to include the cells $(i-2 : i+1, j-2 : j+1)$. Finally the cells $(i-2 : i+1, j-2)$ and $(i-2, j-2 : j+1)$ are eliminated based on the relative distance of their outer nodes to the point P compared to the cells $(i-2 : i+1, j+1)$ and $(i+1, j-2 : j+1)$, respectively. At the end of this process, it is determined that the front point P resides in the domain composed by the cells $(i-1 : i+1, j-1 : j+1)$. The front properties evaluated at the point P are distributed onto these cells and flow variables such

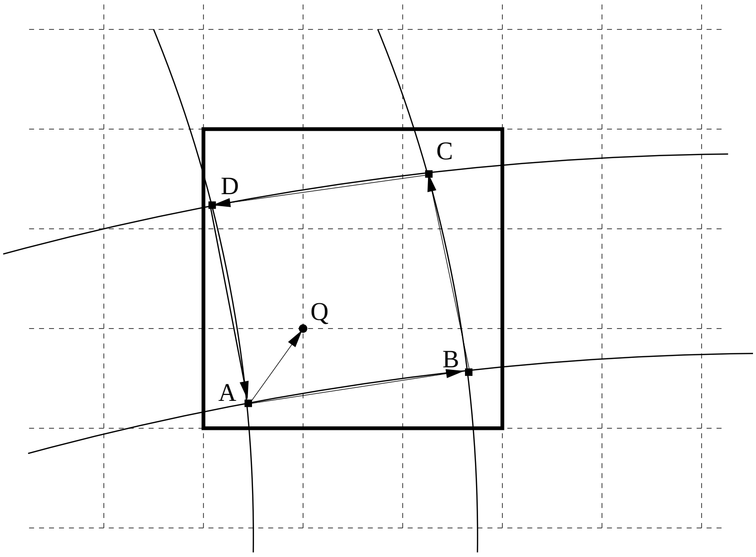


Figure 2.4: Preprocessing of the uniform Cartesian grid. Vector algebra is used to determine which uniform Cartesian grid nodes reside in each curvilinear grid cell [1].

as velocity field are interpolated onto point P from these cells.

2.4 Smoothing the Front Properties onto the Curvilinear Grid

Since the flow equations are solved on the curvilinear grid but the surface tension is computed on the front, it is necessary to convert the surface tension into a body force by an appropriate distribution function [13]. This involves an approximation to the delta function on the curvilinear grid in a conservative manner. Let ϕ_f be an interface quantity per unit surface area, it should be converted into the grid value ϕ_g given per unit volume. To ensure that total value is conserved in the smoothing, we must have

$$\int_{\Delta s} \phi_f(s)ds = \int_{\Delta v} \phi_g(\mathbf{x})dv, \tag{2.45}$$

where Δv is the volume of the grid cell. Following Tryggvason et al. [13], this consistency condition is satisfied by writing

$$\phi_{ij} = \sum_l \phi_l w_{ij}^l \frac{\Delta s_l}{v_{ij}}, \tag{2.46}$$

where ϕ_l is a discrete approximation to the front value ϕ_f, ϕ_{ij} is an approximation to the grid value ϕ_g, Δs is the area of front element l and w_{ij}^l is the weight of grid point ij with

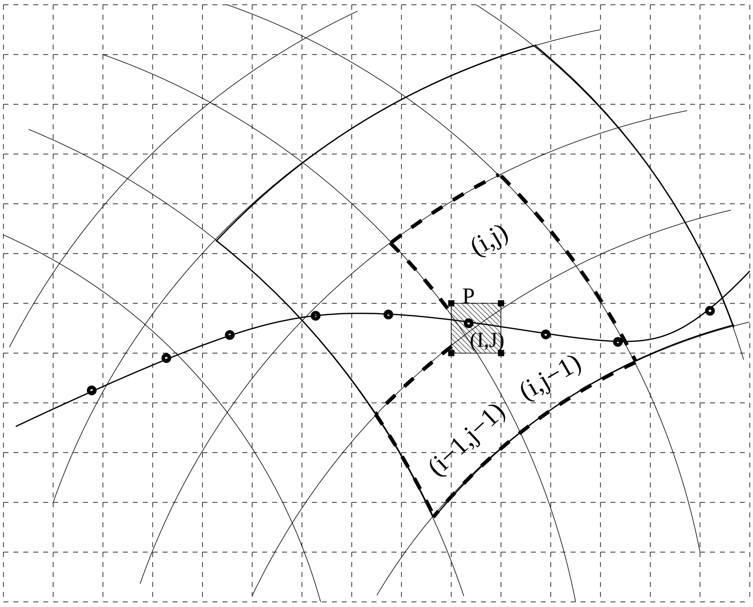


Figure 2.5: The tracking algorithm for curvilinear grids [1].

respect to element l. For consistency, the weights must satisfy

$$\sum_{ij} w_{ij}^{l} = 1, \tag{2.47}$$

but can be selected in different ways [13]. In the present study, the weight for the grid point ij for smoothing from $\mathbf{x}_p = (r_p, z_p)$ is defined as a tensor-product kernel in the form

$$w_{ij}(\mathbf{x}_p) = K(r_r)K(r_z), \tag{2.48}$$

where $r_r = |r_p - r_{ij}|/r_{\max}$ and $r_z = |z_p - z_{ij}|/z_{\max}$. Note that $r_{\max}$ and $z_{\max}$ are the maximum distance of the grid nodes on which the front quantity ϕ_{ij} is to be distributed in r and z directions, respectively. The functional form of K used here is

$$K(\hat{r}) = \begin{cases} \frac{4}{3} - 8(1-\hat{r})\hat{r}^2 & \text{if} \quad \hat{r} \leq 0.5 \\ \frac{8}{3}(1-\hat{r})^3 & \text{if} \quad 0.5 < \hat{r} \leq 1.0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.49}$$

which is symmetric about $\hat{r} = 0$, and piecewise cubic with continuous first and second derivatives. The weights w_{ij} are normalized to satisfy the consistency condition given by Eq. (2.47). The weights w_{ij} are also used to interpolate grid values such as velocity field from the curvilinear grid onto the front points.

2.5 Computation of Material Properties

The material properties such as density and viscosity are computed according to

$$\begin{aligned} \rho &= \phi\rho_d + (1-\phi)\rho_o \\ \mu &= \phi\mu_d + (1-\phi)\mu_o, \end{aligned} \tag{2.50}$$

where the subscripts o and d refer to the ambient and the drop fluids, respectively. The indicator function ϕ is defined such that it is unity inside and zero outside of the drops. If the material properties change suddenly from one grid point to the next grid point disturbances may occur. To prevent these disturbances, the interface is not kept completely sharp but given a small thickness of the order of the grid size. In this transition area (Fig. (2.6)) the fluid properties change smoothly from the value outside the front to the value inside the front. This transition region also helps to determine the position of the interface more accurately on the grid. Figure (2.7) shows the smooth change of the indicator on the curvilinear grid for the axisymmetric problem of bouyancy-driven drop in a straight channel. Following Tryggvason et al. [13], the indicator function is obtained by solving the Poisson equation

$$\nabla^2\phi = \nabla_h \cdot \nabla_h\phi, \tag{2.51}$$

where ∇_h is the discrete version of the gradient operator. The jump $\nabla_h\phi$ is distributed on the neighboring grid cells using the Peskin distribution [13] and Eq. (2.51) is then solved on the uniform grid in the vicinity of each drop. After computing the indicator function on the uniform grid, it is interpolated onto the curvilinear grid using bilinear interpolations.

2.6 Surface Tension

The surface tension on each front element is computed following the procedure described by Tryggvason et al. [13]. The surface tension on a small front element can be computed as

$$\delta F_\sigma = \int_{\Delta s} r\sigma\kappa\mathbf{n}ds. \tag{2.52}$$

Using the definition of curvature of a two-dimensional curve, i.e., $\kappa\mathbf{n} = \partial\mathbf{s}/\partial s$ and accounting for the axisymmetry of the problem, Eq. (2.52) can be integrated to yield

$$\delta F_\sigma = r\sigma(\mathbf{s}_2 - \mathbf{s}_1) - \Delta s\sigma\mathbf{e}_r, \tag{2.53}$$

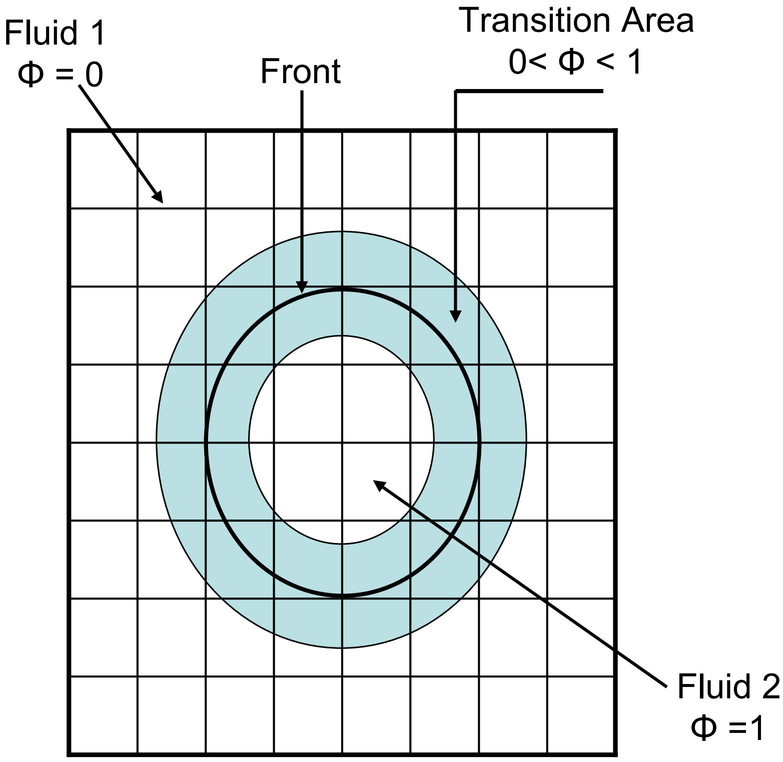


Figure 2.6: Transition region from one phase to another using indicator function.

where $\mathbf{e}_r$ denotes the unit vector in the radial direction. The tangent vector to the curve $\mathbf{s}$ is computed directly by a Lagrange polynomial fit through the end-points of each element and the end-points of adjacent element in the same way as described by Tryggvason et al. [13]. Then, the distributed surface tension force is added as a body-force-like term into the equation of motion.

2.7 The Overall Solution Procedure

The finite-volume and front-tracking methods described above are combined as follows. In advancing solutions from physical time level n ($t_n = n \cdot \Delta t$) to level $n+1$, the locations of the marker points at the new time level $n+1$ are first predicted using an explicit Euler

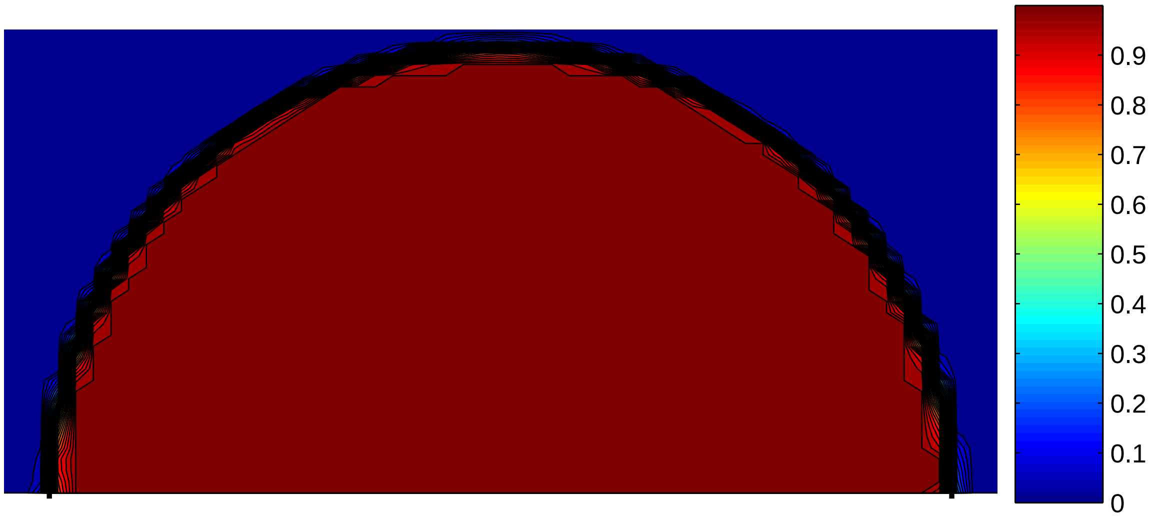


Figure 2.7: Transition region from one phase to another using indicator function on the curvilinear grid.

method, i.e.,

$$\widetilde{\mathbf{X}}_p^{n+1} = \mathbf{X}_p^n + \Delta t \mathbf{V}_p^n, \tag{2.54}$$

where $\mathbf{X}_p$ and $\mathbf{V}_p$ denote the position of front marker points and the velocity interpolated from the neighboring curvilinear grid points onto the front point $\mathbf{X}_p$, respectively. Then the material properties and surface tension are evaluated using the predicted front position as

$$\rho^{n+1} = \rho(\widetilde{\mathbf{X}}_p^{n+1}); \quad \mu^{n+1} = \mu(\widetilde{\mathbf{X}}_p^{n+1}); \quad \mathbf{f}_b^{n+1} = \mathbf{f}_b(\widetilde{\mathbf{X}}_p^{n+1}). \tag{2.55}$$

The velocity and pressure fields at new physical time level $n+1$ are then computed by solving the flow equations by the FV method for a single physical time step and finally the positions of the front points are corrected as

$$\mathbf{X}_p^{n+1} = \mathbf{X}_p^n + \frac{\Delta t}{2}(\mathbf{V}_p^n + \mathbf{V}_p^{n+1}). \tag{2.56}$$

After this step the material properties and the body forces are re-evaluated using the corrected front position. The method is second order accurate both in time and space.

Perfect reflection boundary conditions are used at the solid boundary for the front marker points, i.e., the front marker points crossing the solid boundary due to numerical error are

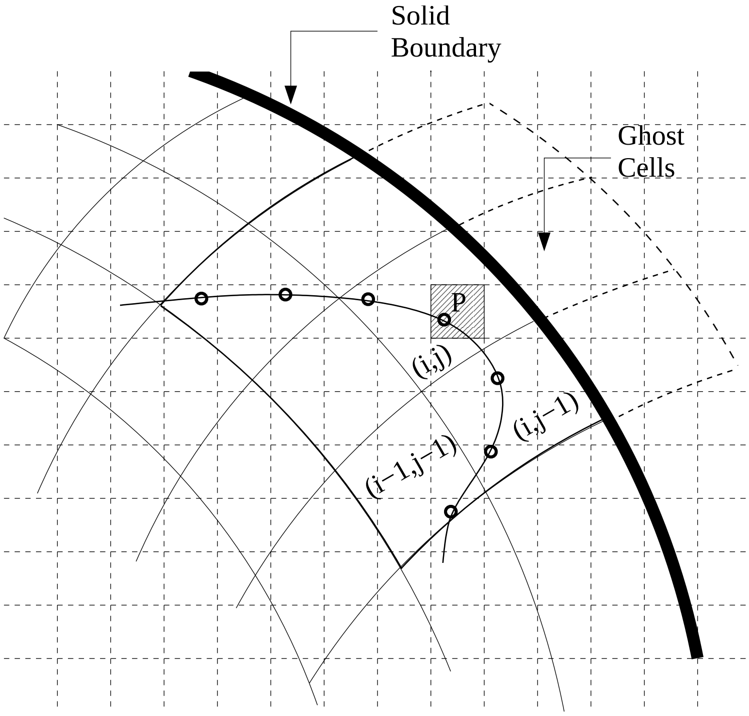


Figure 2.8: Distribution of front properties from the Lagrangian grid onto the curvilinear grid and interpolation of flow quantities from the curvilinear grid onto the Lagrangian grid near a solid boundary [1].

reflected with respect to the inward normal vector back into the computational domain. If the front marker point is close to the boundary as sketched in Fig. (2.8), the front properties are distributed onto curvilinear cells in a conservative manner, i.e., the weights are defined only for the cells within the computational domain (shown by the thick solid line in the sketch) and are normalized to satisfy the consistency conditions given by Eq. (2.47). The grid properties are interpolated onto the front point in a similar manner. The Lagrangian grid is initially uniform and is kept nearly uniform throughout the computations by deleting small elements and splitting the large elements in the same way as described by Tryggvason et al. [13]. The initial front element size is typically set to $0.75\Delta l$, where Δl is the minimum size of the curvilinear grid cell. During the simulation, in each physical time step, the elements that are smaller than $0.5\Delta l$ are deleted and the elements that are larger than Δl are splited into two parts in order to keep the Lagrangian grid nearly uniform and to prevent the formation of wiggles much smaller than the grid size.

Chapter 3

RESULTS AND DISCUSSION

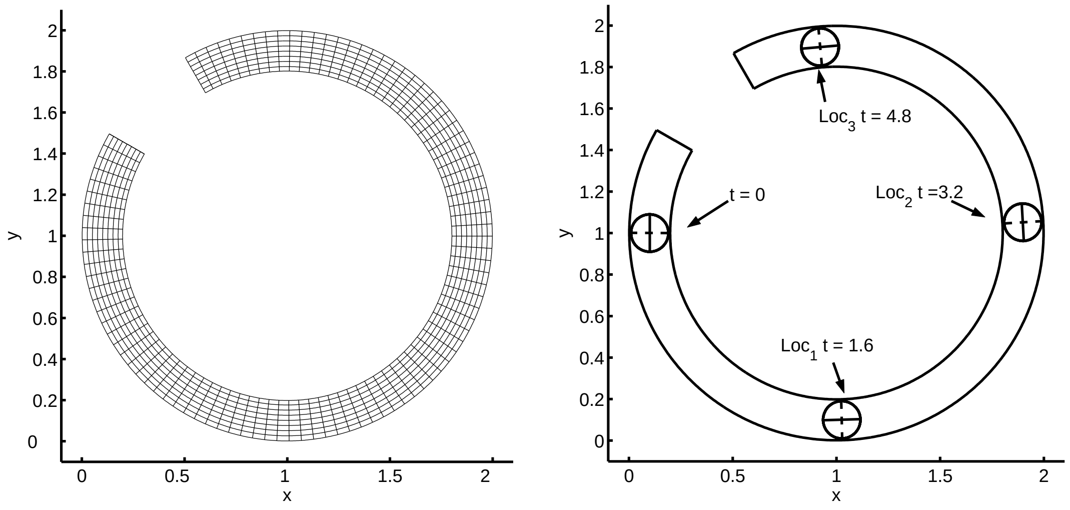
First the particle tracking algorithm is tested in a simple setting of a rigid body rotation of fluid in a circular channel. The method is then applied to compute the motion of a freely falling drop in a straight channel studied earlier by Han and Tryggvason [2] using finite-difference/front-tracking (FD/FT) method. Finally the method is used to compute the film thickness deposited behind a gas phase steadily displacing a newtonian fluid and the results are compared with the theoretical result of Bretherton [26] and Taylor [25], and with the computational results obtained earlier by Ratulowski and Chang [29], Giavedoni and Saita [28] and with the experimental results obtained by Aussillous and Qu*éré* [27].

3.1 Particle tracking

First the particle tracking algorithm is tested in a simple setting of a rigid body rotation of fluid in a circular channel as shown in Fig. (3.1). This test case is used by Muradoglu and Kayaalp [1] to demonstrate the temporal and spatial accuracies of the tracking method. The radius of the outer boundary and the width of the channel are set to $R_c = 1$ and $w_c = 0.2$ respectively. The velocity field is specified as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -y + y_0 \\ x - x_0 \end{pmatrix}, \tag{3.1}$$

where x and y are the components of the two-dimensional coordinates; x_0 and y_0 are the centroid of the circular channel; and u and v are the x and y components of the velocity vector, respectively. A two dimensional drop of diameter $d_d = 0.15$ centered at $(x, y) = (0.1, 1.0)$ is set into motion by the fluid and two equatorial lines are used to visualize its motion (Fig. (3.1b)). A coarse version of the curvilinear grid used in the simulations is shown in Fig. (3.1a). To demonstrate the accuracy of the tracking algorithm, snapshots of the drop at the initial and three other different locations in the channel are shown in Fig. (3.1b). This figure clearly shows the rigid body motion of the drop indicating the



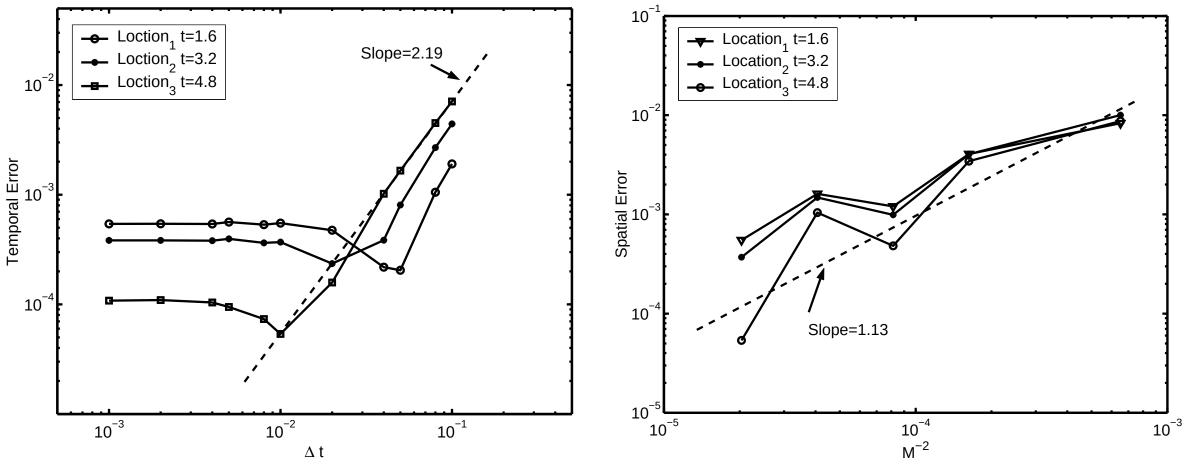
(a) (b)

Figure 3.1: (a) A coarse curvilinear grid containing 96×8 grid cells. (b) The evolution of a two-dimensional drop moving with a fluid in a rigid body rotation. (Grid: 768×64, $dt = 10^{-3}$)

accuracy of the tracking algorithm. The temporal and spatial accuracies of the tracking algorithm are quantified in Fig. (3.2a) and Fig. (3.2b), respectively. The error is defined as the difference between the computed and exact locations of the drop centroid and is plotted at three different locations $t = 1.6$, $t = 3.2$ and $t = 4.8$. In Fig. (3.2a), the error is plotted against the time step Δt. This figure clearly shows that the tracking algorithm is second order accurate in time. It can be also seen that for very small time steps the error approches a constant value as expected since the temporal error becomes negligible compared to the spatial error for time step Δt smaller than a specific value. In Fig. (3.2b), the error is plotted against the inverse of the total number of grid cells M^{-2}. As can be seen in this figure, the tracking algorithm is also second order accurate in space as expected.

3.2 Bouyancy-Driven Falling Drop in a Straight Channel

The second test case concerns with the bouyancy-driven falling drops in a straight channel studied earlier by Han and Tryggvason [2] and by Muradoglu and Kayaalp [1]. The physical problem and computational domain are sketcehd in Fig. (3.3b).



(a) (b)

Figure 3.2: Error in the position of the drop centroid (a) against the time step Δt and (b) against the inverse of the total number of grid cells M^{-2}.

As can be seen in this figure, the ambient fluid completely fills the rigid cylinder and the drop that is denser than the ambient fluid accelerates downward due to gravitational body force. The problem is governed by four nondimensional parameters [2], namely the Eötvös number Eo (interchangeably called the Bond number, Bo), the Ohnesorge number Oh_d, the density and viscosity ratios defined as

$$\begin{aligned} E_o &= \frac{g_z \Delta\rho d^2}{\sigma}, \\ Oh_d &= \frac{\mu_d}{\sqrt{\rho_d d\sigma}}, \\ \rho^* &= \frac{\rho_d}{\rho_o}, \\ \mu^* &= \frac{\mu_d}{\mu_o}, \end{aligned} \qquad (3.2)$$

where $\Delta\rho = \rho_d - \rho_o$ is the density difference between the drop and the ambient fluids, g_z is gravitational acceleration and d is the initial drop diameter. The Ohnesorge number based on the ambient fluid is defined similarly as $Oh_o = \frac{\mu_o}{\sqrt{\rho_o d\sigma}}$. The subscripts d and o denote the properties of the drop and ambient fluids, respectively. The nondimensional time is defined as

$$t^* = \frac{t}{\sqrt{d/g_z}}. \qquad (3.3)$$

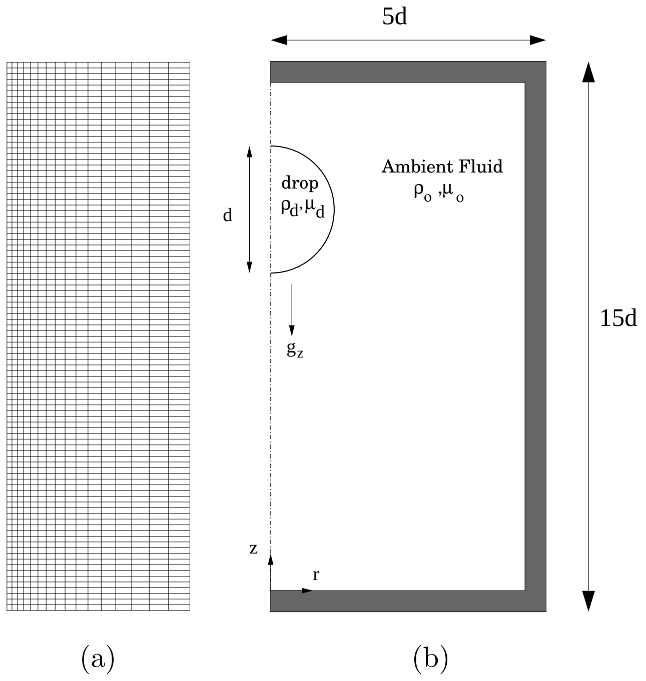


Figure 3.3: a) A coarse computational grid containing 16x96 cells. The grid is stretched in radial direction. b) Schematic illustration of the physical problem and computational domain for a buoyancy-driven falling drop in a straight channel.

In all the computations presented in this section, the computational domain is $5d$ in radial direction and is $15d$ in the axial direction. No-slip boundary conditions are applied on the cylinder walls and axisymmetry conditions are applied on the centerline. The drop centroid is initially located at $(r_c, z_c) = (0, 12d)$. The computational domain is resolved by a 128×768 regular Cartesian grid. A coarse grid containing 16x96 grid cells is plotted in Fig. (3.3a) to show the overall structure of the grid used in the simulations. The grid is stretched in the radial direction to have more grid points close to the centerline. The Ohnesorge number, the density and viscosity ratios are kept constant at $Oh_d = 0.0466$ ($Oh_o = 0.05$), $\rho_d/\rho_o = 1.15$ and $\mu_d/\mu_o = 1$ in all the results presented here.

First, the evolution of the drop for various Eo numbers ($Eo = 6$, $Eo = 12$ and $Eo = 24$) is shown in Fig. (3.4). Then, the results of the present FV/FT method are compared with the finite-difference/front-tracking FD/FT results obtained earlier by Han and Tryggvason [2]. To quantify the accuracy of the present method, the velocity is nondimensinalized by $\sqrt{g_z d}$. The nondimensional velocity of the drop centroid and the percentage change in the drop

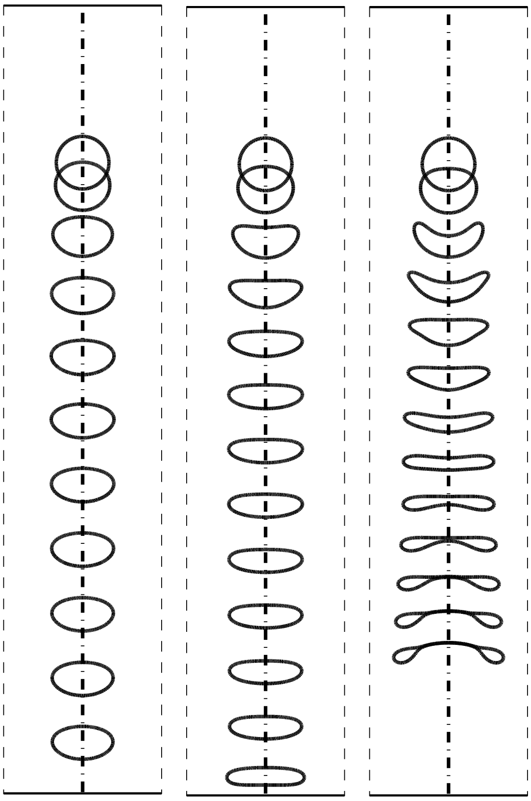
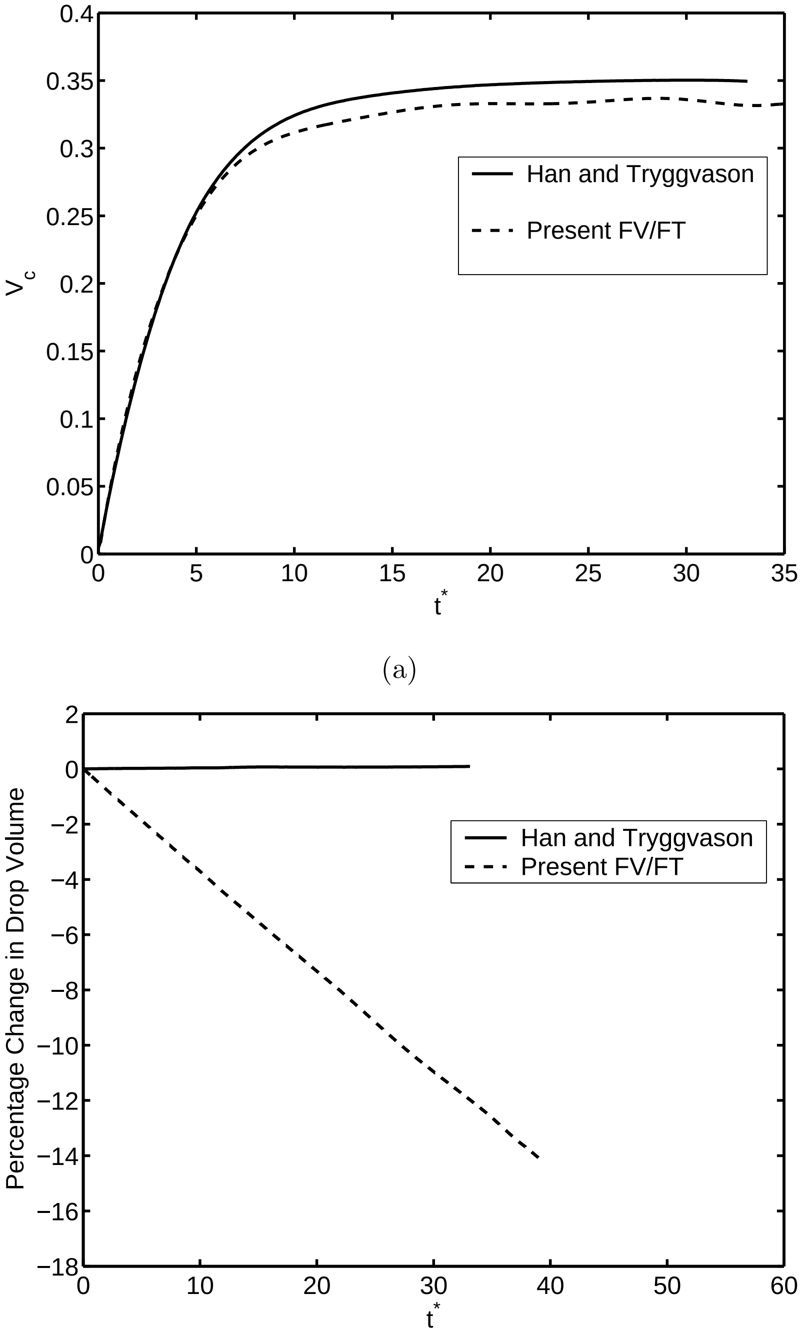


Figure 3.4: Evolution of drop for $Eo = 6$ (left), $Eo = 12$ (center) and $Eo = 24$ (right). The gap between two successive drops in each column represents the distance the drop travels at a fixed time interval and the last interface is plotted at $t^* = 44.04$, $t^* = 44.12$ and $t^* = 43.92$ for $Eo = 6$, $Eo = 12$ and $Eo = 24$ cases, respectively.

volume for ($Eo = 6$, $Eo = 12$ and $Eo = 24$) are plotted together with the results of Han and Tryggvason in Figs. (3.5), (3.6) and (3.7), respectively.

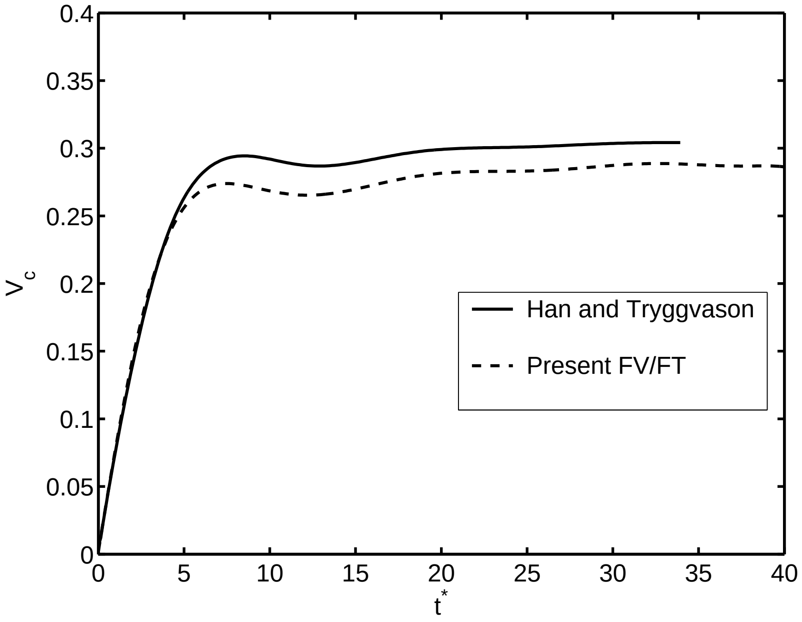
It can be seen that the percentage change in drop volume is relatively large in the present FV/FT method compared to FD/FT method of Han and Tryggvason [2]. This relatively large volume change in the present method may be attributed to the large numerical errors in the present results due to larger physical time steps, interpolation and distribution algorithms and partly due the front restructuring algorithm. The nondimensinal velocity in the present method is found to be in very good qualitative agreement with the early results but quantitavely it is smaller than previously obtained results and this is due to the relatively large volume loss in the present work.



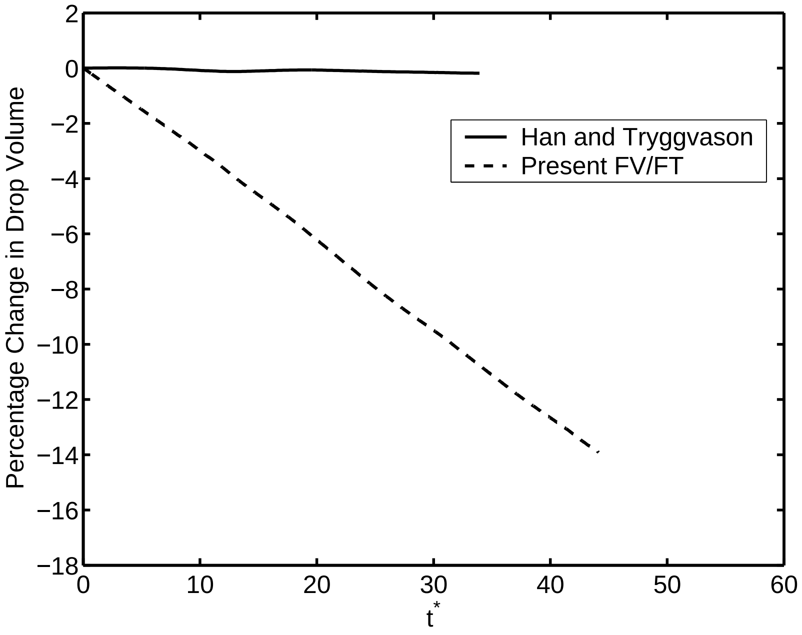
(a)

(b)

Figure 3.5: a) Velocity of drop centroid and b) percentage change in drop volume versus t^* for $Eo = 6$. The dashed lines are the present results and solid lines are Han and Tryggavason FD/FT results [2].

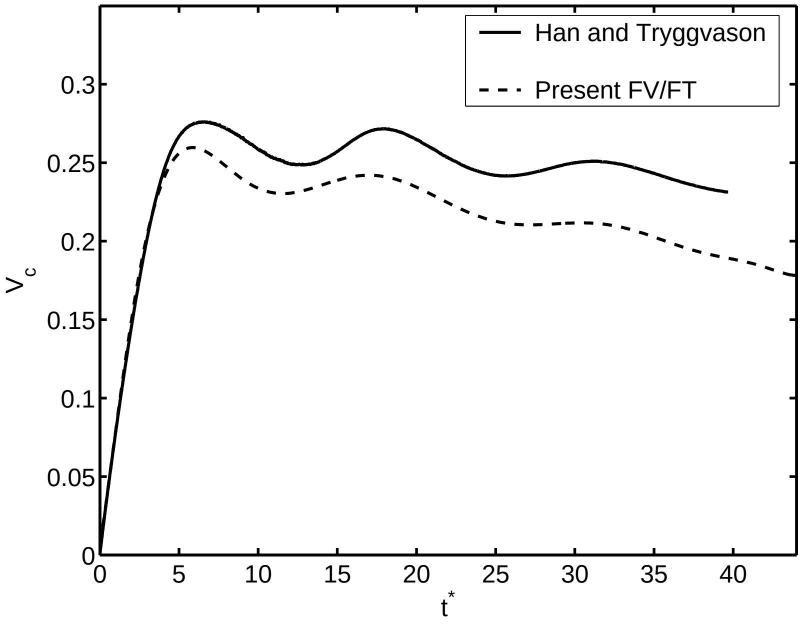


(a)

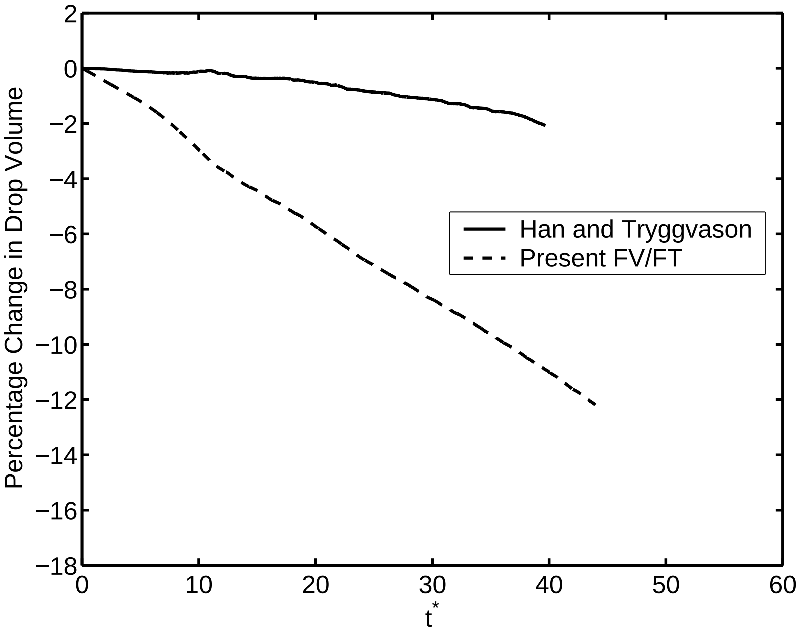


(b)

Figure 3.6: a) Velocity of drop centroid and b) percentage change in drop volume versus t^* for $Eo = 12$. The dashed lines are the present results and solid lines are Han and Tryggavason FD/FT results [2].



(a)



(b)

Figure 3.7: a) Velocity of drop centroid and b) percentage change in drop volume versus t^* for $Eo = 24$. The dashed lines are the present results and solid lines are Han and Tryggavason FD/FT results [2].

3.3 Deposition of a fluid on the wall of a tube

If a drop of a wetting liquid is displaced by air in a tube, the tube remains wet behind the drop as shown in Fig. (3.8). A very classical (and practical) problem is the knowledge of liquid remaining on the wall of the tube, namely h. The problem has been extensively studied in the limit of low velocities of deposition, in particular, since Taylor's [25] and Bretherton's [26] first experiments. The problem has also been studied by Aussillous and Qu*éré* [27] at high deposition speeds with liquids of low viscosity stressing the influence of inertia on the film thickness. After a brief summary of the these regimes, the new FV/FT method is validated for the same deposition problem and the results are compared with those obtained in early experiments.

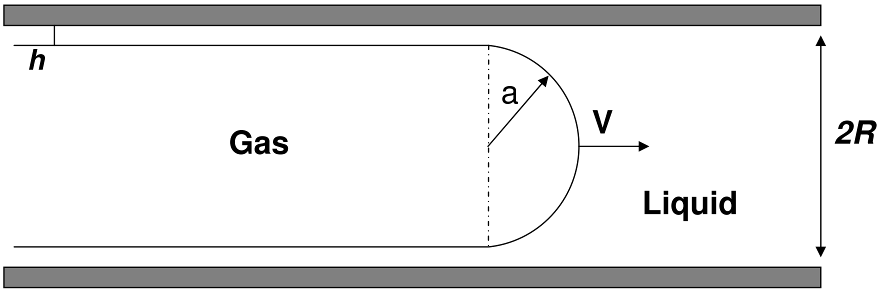
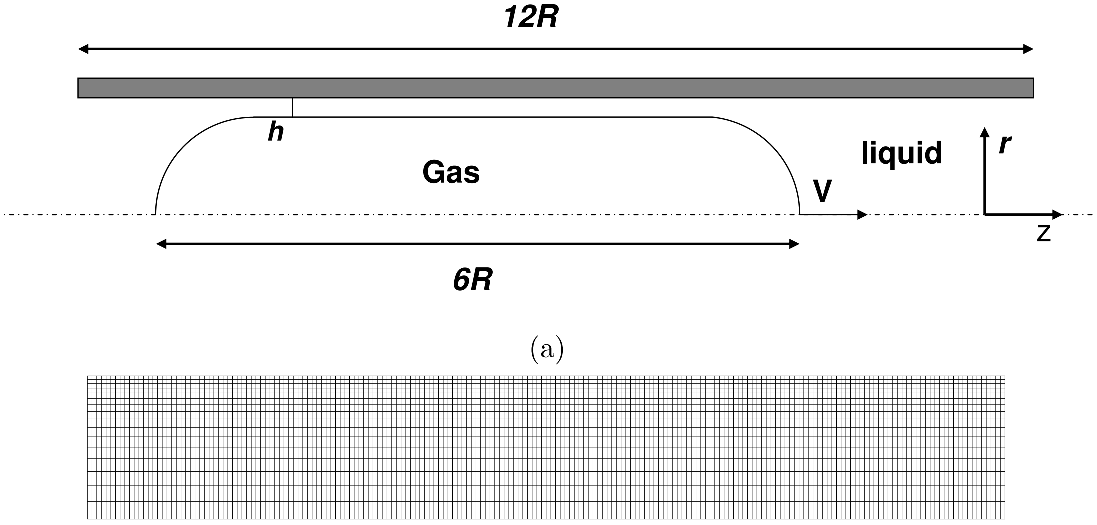


Figure 3.8: A drop of a wetting liquid moved in capillary tube leaves behind a film. The thickness h of this film generally depends on the drop velocity V. The tube radius is denoted by R.

In the limit of slow deposition, the boundary condition at the *solid-liquid interface* causes the deposition of a film. Since viscous and capillary forces play antagonist roles, the nondimensional parameter to be considered is the ratio of these forces, the so called *capillary number* $Ca = \mu V/\sigma$, where μ, V and σ are the liquid viscosity, the bubble velocity and the surface tension, respectively. The regime where the film thickness only depends on Ca is called *visco-capillary*. In this regime, the film thickness was calculated by Bretherton using planar lubrication equations [26]. The flow takes place because of the Laplace pressure difference ($\Delta p = \sigma/a$), where a is the radius of curvature of the spherical bubble, due to the difference in curvature between the film and the meniscus. Balancing the viscous forces



(a)

(b)

Figure 3.9: (a) The computational domain. (b) A coarse computational grid containing 96x16 cells. The grid is stretched in radial direction.

with the pressure gradient yields

$$\frac{\sigma V}{h^2} \sim \frac{1}{\lambda}\frac{\sigma}{R} \tag{3.4}$$

where the film length λ is calculated by balancing the Laplace pressures:

$$-\frac{\sigma}{r} - \frac{\sigma h}{\lambda^2} \sim -\frac{2\sigma}{R}. \tag{3.5}$$

Thus we have $\lambda \sim \sqrt{hR}$, from which the classical *Bretherton law* can be deduced [26]:

$$\frac{h}{R} \sim Ca^{2/3}. \tag{3.6}$$

This equation is set for small capillary numbers assuming negligible inertial effects ($h << R$ implies $Ca << 1$). As Ca increases, R must be replaced by $(R - h)$ in Eq. (3.4) and Eq. (3.5) so that the deposition scaling law becomes

$$\frac{h}{R} \sim \frac{Ca^{2/3}}{1 + Ca^{2/3}} \tag{3.7}$$

The above equation implies a dependence of the normalized film thickness on one single parameter, namely the capillay number and a convergence of the thickness at large Ca, obviously related to the confinement which imposes $h < R$ [27]. These results agree with the data obtained by Taylor with different viscous oils [25]. A fit of the form $h/R \sim 1.34Ca^{2/3}/\left(1 + 1.35 * 2.5Ca^{2/3}\right)$ describe quite precisely the data, where the coefficient 1.34 was derived by Bretherton; the coefficient 2.5 is empirical. This convinient empirical equation is called *Taylor's law*. Excellent numerical solutions of the problem, which remarkably fit the Taylor's data, have been proposed by, in particular by Reinelt and Saffman [30], Martinez and Udell [31] and Giavedoni and Saita [28].

The problem is simulated assuming a bubble of initial length ($\sim 6R$) located in a channel of length and diameter ($12R$) and ($2R$), respectively. The computational domain of the axisymmetric problem used in the simulation is sketched in Fig. (3.9a). The computational domain is resolved by 1280x96 uniform Cartesian grid. The grid is stretched in the radial direction to have more grid points close to the wall of the channel. In this study, different series of computations are performed to compute the film thickness in both low speed and high speed deposition regions. Two sets of computations are perform in the low speed deposition region with capillary number in the range of $Ca = 0.005 - 1.0$. The first set of computations is performed assuming viscous oils ($\mu \sim 10^{-1}Pa \cdot s$) and the second set is computed assuming less viscous liquids ($\mu \sim 10^{-3}Pa \cdot s$, the water viscosity). The computations are performed for Reynolds number on the order of $Re \sim (10^{-1})$ and $Re \sim (1)$ for the first and second series, respectively. The physical time step used in both series of computations is on the order of $\Delta t \sim (10^{-4})$. The data obtained for both sets are displayed together with the emperical fit of Taylor Law in Fig. (3.10). It can be seen that results are in a good agreement with the Taylor law.

In the high speed deposition region the thickening effect of inertia, studied experimentally by Aussillous and Qu*éré* [27], is investigated. Aussillous and Qu*éré* studied the film thickness deposed by drops of ethanol moved in a capillary tube of radius $R = 0.78mm$. They showed that the film thickness only obeys Taylor's law at small capillary number [27]. In the present work, two series of computations are performed to study the thickenin effect of inertia. These computations are done for the same range of capillary numbers as in the experiment performed by Aussillous and Qu*éré* and the results are plotted together with Taylor law in Fig. (3.11) and Fig. (3.12). The results obtained in the computations showed a small shift from Taylor law indicating the thickening effect for the whole range studied of capillary number. This is attributed to the limited value of density and viscosity ratio used

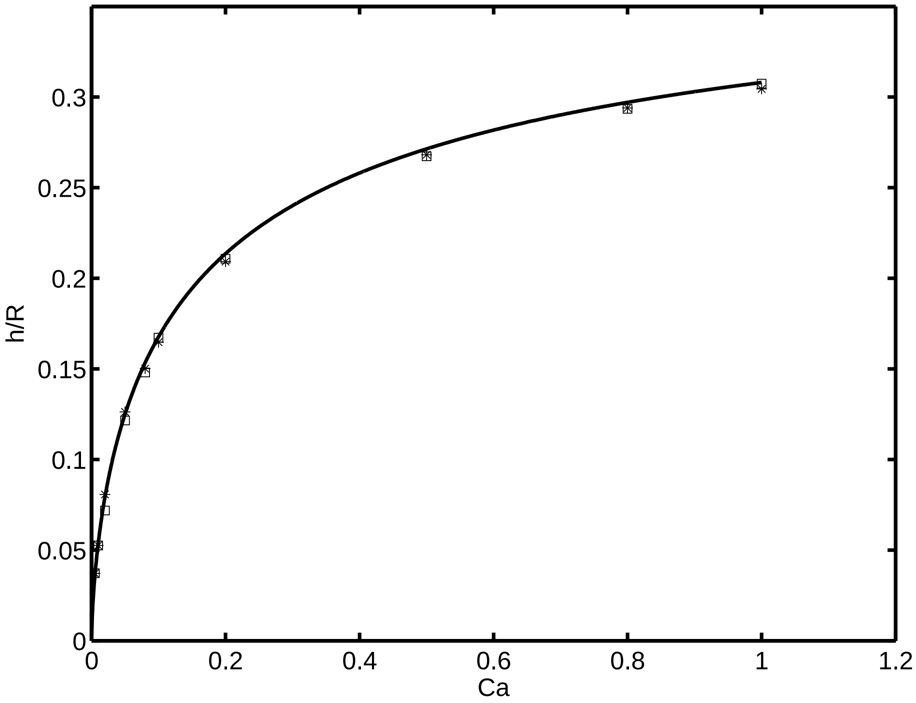


Figure 3.10: Normalized film thickness as a function of the capillary number for different liquids. The full line is the empirical fit of Taylor's law. Asterisks and open squares are the results obtained assuming viscous oils ($\mu \sim 10^{-1} Pas$, $Re \sim (10^{-1})$) and less viscous liquids ($\mu \sim 10^{-3} Pas$, the water viscosity, $Re \sim (1)$), respectively. These results correspond to low speed deposition.

in the simulation. In this work, interface profile is also studied. At low Ca, Ratulowski and Chang [29] showed that the front and rear regions of the interface profile are nearly static profiles of spherical caps for isolated bubbles with undulating rear profile and of nodoidal surfaces for bubble trains. Near the channel wall is a region where both viscous and capillary forces are important. This region is called the transition region. For very long bubbles, the transition region merges into a region of uniform film thickness. This uniform film region is not present for short bubbles [29]. In Fig. (3.13), the front and back bubble profiles for very long bubble at various Ca values are displayed. The monotonic front profile Fig.(3.13a) and the formation of the undulating back profile Fig.(3.13b) for relatively high capillary numbers can be observed.

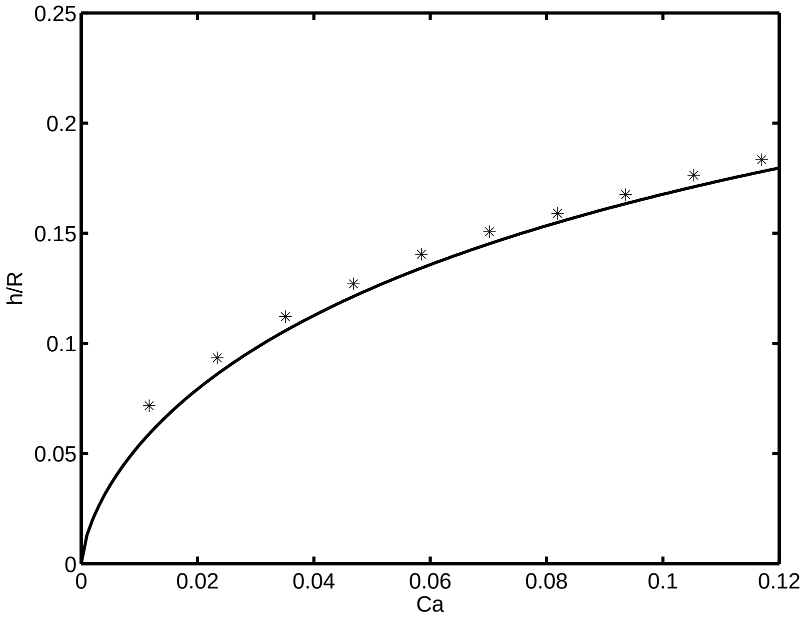


Figure 3.11: Normalized film thickness as a function of the capillary number. The full line is the empirical fit of Taylor's law. Asterisks are the results computed assuming Reynolds number is of order $Re \sim 10^2$.

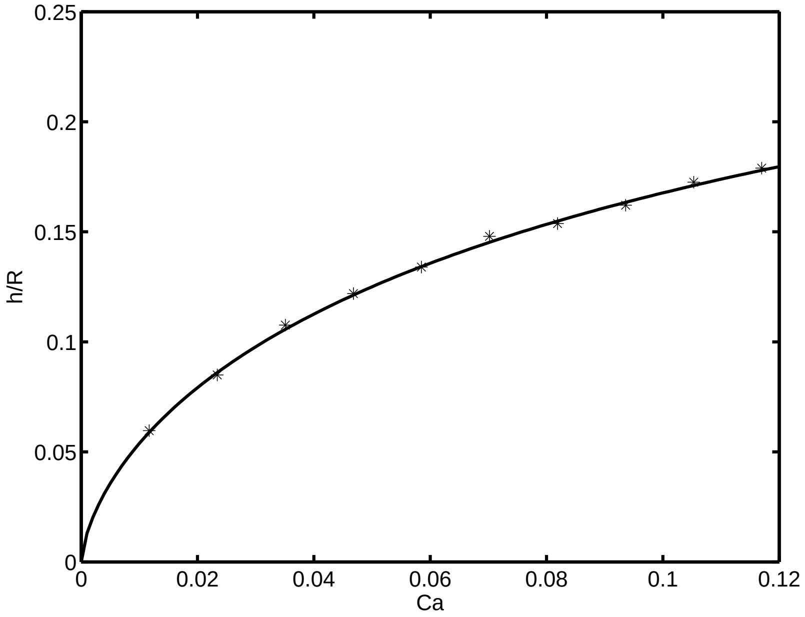
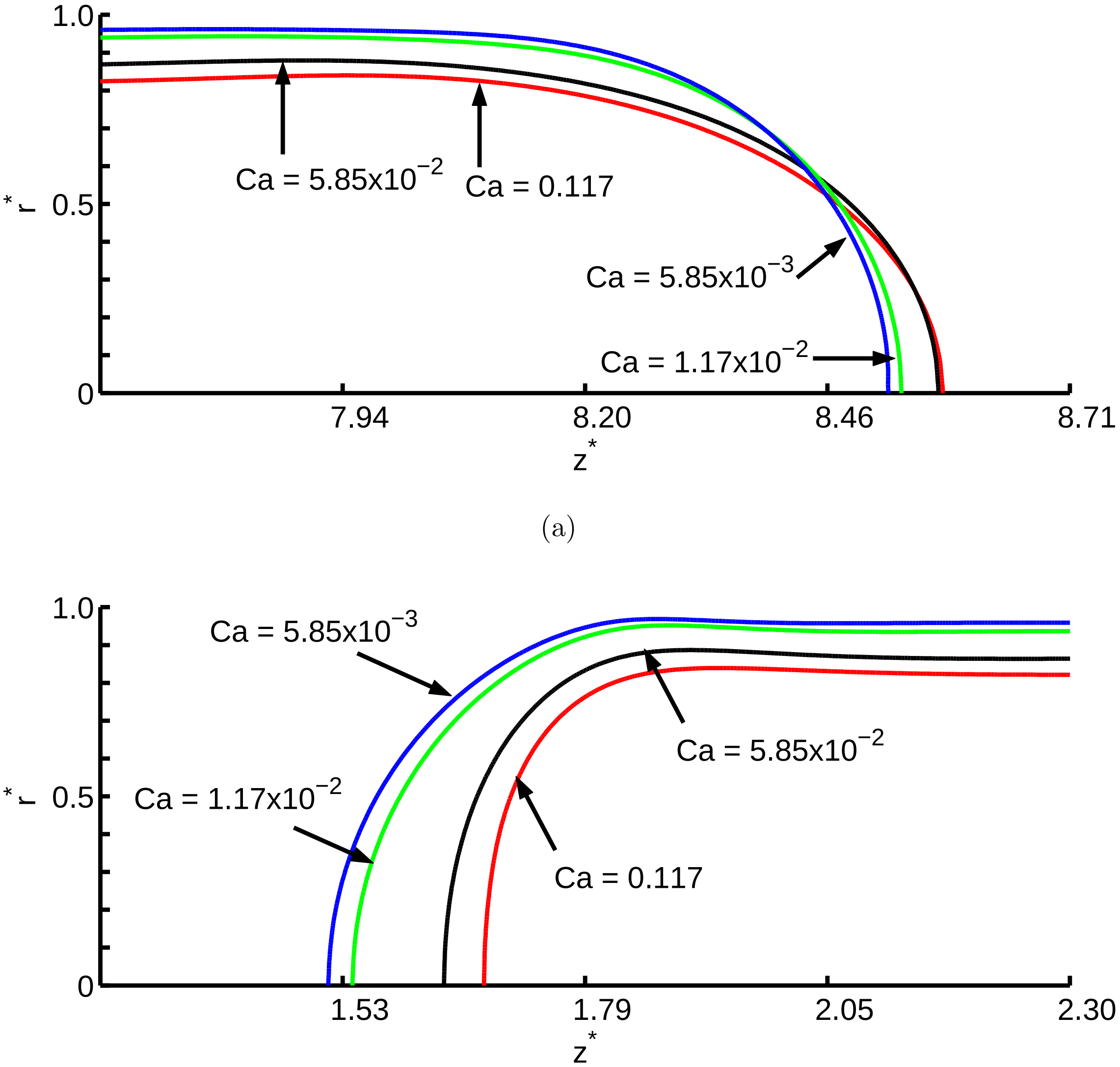


Figure 3.12: Normalized film thickness as a function of the capillary number. The full line is the empirical fit of Taylor's law. Asterisks are the results computed assuming Reynolds number is of order $Re \sim 10^3$.



(a)

(b)

Figure 3.13: (a) Front (b) Back film profile for very long bubbles at various capillary numbers. The monotonic front profile and the formation of undulating back profile are evident. r and z are the nondimensional radial and axial components of front location, respectively.

Chapter 4

CONCLUSIONS

A finite-volume/front-tracking FV/FT method has been developed for computations of dispersed multiphase flows in complex geometries. The method is based on the one-field formulation of the flow equations and treating the different phases as a single fluid with variable material properties. The flow equations are solved by a SIMPLE finite volume method on a body-fitted curvilinear grid and a separate Lagrangian grid is used to represent the interfaces between different phases. A novel tracking algorithm that utilizes an auxiliary uniform Cartesian grid is adopted to track the interface on the curvilinear grid and is found to be robust and computationally efficient. The front-tracking methodology is extended to be used with the complex grid generator associated with the FV solver code namely *caffa* [24] and to facilitate accurate and efficient modeling of strong interactions between the phases and complex solid boundaries.

The FV method is based on **S**emi-**I**mplicit-**P**ressure-**L**inked-**E**quation (SIMPLE) algorithm. In this method, a pressure-correction equation is used to enforce mass conservation at each time step. Being an implicit scheme, the SIMPLE algorithm is prefered for steady and slow transient flows, because it has less stringent time step restrictions than explicit scheme. The numerical method including discritization schemes as well as approximation of all terms are explained in details in chapter 2.

The method is implemented to solve two-dimensional (plane or axisymmetric) dispersed multiphase flows and has been successfully applied to several test cases. First, the particle tracking algorithm of the new finite-volume/front-tracking FV/FT method is validated and the temporal and spatial errors are quantified for the rigid body motion case demonstrating that the method is convergent in terms of grid refinement and time stepping error. Then, the new FV/FT method is applied to compute the motion of a bouyancy-driven falling drop in a straight channel and the results are in a very good qualitative agreement with those of the FD/FT method developed by Han and Tryggvason [2]demonstrating the accuracy of the present method. The relatively large volume change of the drop in the present method is attributed to the large numerical error in the present results due to larger physical time steps, interpolation and distribution algorithms and partly due the front restructuring algorithm. The differnce in nondimensional velocity between the results of the present work and the results of Han and Tryggvason is attributed to the relatively large volume change of the drop in the present work.

Further work includes the study of transport of gas bubbles in capillaries and the computation of the film thickness of fluid left behind a drop moved inside a capillary tube. This problem is studied experimentally by Bretherton and Taylor [25, 26] and by Aussillous and Qu*é*r*é* [27]. Computational results are also obtained earlier by Reinelt and Saffman [30], Martinez and Udell [31] and Giavedoni and Saita [28]. In this study, series of computations corresponding to various capillary numbers are performed demonstrating the low and high speed deposition results. Computations performed in the low speed deposition region are in very good qualitative and quantitative agreement with the emperical fit of Taylor Law demonstrating that the film thickness obeys Taylow law in this region. Aussillous and Qu*é*r*é* experimentally studied the thickening effect of inertia and showed that the film thickness only obeys Taylor's law at small capillary number then the film becomes thicker. In the present work, series of computations are performed to demonstarte this effect. The results obtained showed that the film thickness obeys Taylor's law with a shift indicating thicker films. This may be attributed to the limited density and viscosity ratios used in computations as they become important in the high speed region. Finally, the front and back film profiles for long bubbles are demonstrated and compared with the results of Ratulowski and Chang [29]. The front and rear of the interface have spherical profiles and the formation of the undulating back profile is observed.

Future work includes the study of the surfactant and its effects on two-phase flows. Break-up of drops moving in contricted capillaries studied experimentally by Hemmat and Borhan [9] are also included in the future work. Simulations of biofluid dynamics is also possible if further improvement of the FV/FT method is achieved under very low Reynolds numbers. The method can be then applied to simulations of white and red blood cells through bifurcating capillaries in the circulation system. In particular, adhesion dynamics of leukocytes to ligand-coated surfaces can be modeled and simulated within the framework of finite-volume/front-tracking FV/FT methodology.

BIBLIOGRAPHY

[1] M. Muradoglu and A. D. Kayaalp, "An auxiliary grid method for computations of multiphase flows in comples geometries," *J. Comput. Phys.*, 2006.

[2] J. Han and G. Tryggvason, "Secondary breakup of axisymmetric liquid drops: I. accelerartion by a constant body force," *Phys. Fluids*, vol. 11(12), pp. 3650–3667, 1999.

[3] H. A. Stone, A. D. Stroock, and A. Ajdary, "Engineering flows in small devices: Microfluidics towards lab-on-a-chip," *Annu. Rev. Fluid Mech*, vol. 36, pp. 381–411, 2004.

[4] L. Fauci and S. Gueron, "Computational modeling in bilogical fluid dynamics," *Springer-Verlag*, 2001.

[5] T. M. Tsai and M. Miksis, "Dynamics of a droplet in a constricted capillary tube," *J. Fluid Mech.*, vol. 274, pp. 197–217, 1994.

[6] M. Manga, "Dynamics of a droplets in branched tubes," *J. Fluid Mech.*, vol. 315, pp. 105–117, 1996.

[7] H. S. Udaykumar, H. C. Kan, W. Shyy, and R. Tran-Son-Tay, "Multiphase dynamics in arbitrary geometries on fixed cartesian grids," *J. Comput. Phys.*, vol. 137, pp. 366–405, 1997.

[8] W. L. Olbricht and L. G. Leal, "The creeping motion of immiscible drops through a converging/diverging tube," *J. Fluid Mech.*, vol. 134, pp. 329–355, 1983.

[9] M. Hemmat and A. Borhan, "Bouyancy-driven motion of drops and bubbles in a periodically constricted capillary," *Chem. Eng. Commun.*, vol. 150, pp. 363–384, 1996.

[10] S. Osher and R. P. Fedkiw, "Level set methods: An overview," *J. Comput. Phys.*, vol. 169(2), pp. 463–502, 2001.

[11] R. Scardovelli and S. Zaleski, “Direct numerical simulation of free-surface and interfacial flow,” *Annu. Rev. Fluid Mech*, vol. 31, pp. 567–603, 1999.

[12] J. A. Sethian and P. Smereka, “Level set methods for fluid interfaces,” *Annu. Rev. Fluid Mech*, vol. 35, pp. 341–, 2003.

[13] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, and S. N. an Y. J. Jan, “A front-tracking method for the computations of multiphase flow,” *J. Comput. Phys.*, vol. 169(2), pp. 708–759, 2001.

[14] T. Yabe, F. Xiao, and T. Utsumi, “The constricted interpolation profile (cip) method for multiphase analysis,” *J. Comput. Phys.*, vol. 169(2), pp. 556–593, 2001.

[15] C. W. Hirt and B. D. Nichols, “Volume of fliud (vof) method for the dynamics of free boundaries,” *J. Comput. Phys.*, vol. 39, pp. 201–225, 1981.

[16] M. Sussman, P. Smereka, and S. Osher, “A level set approach for computing solutions to incompressible two-phase flows,” *J. Comput. Phys.*, vol. 114, pp. 146–159, 1999.

[17] G. Ryskin and L. G. Leal, “Numerical simulation of free-boundary problems in fluid mechanics, part 2. bouyancy-driven motion of a gas bubble through a quiescent liquid,” *J. Fluid Mech*, vol. 148, pp. 19–35, 1984.

[18] C. W. Hirt, A. A. Amsden, and J. L. cook, “An arbitrary lagrangian-eulerian computing method for all flow speeds,” *J. Comput. Phys.*, vol. 135, pp. 203–216, 1997.

[19] S. O. Unverdi and G. Tryggvason, “A front-tracking method for viscous incompressible multiphase flows,” *J. Comput. Phys.*, vol. 100, pp. 25–37, 1992.

[20] C. Peskin, “Numerical analysis of blood flow in the heart,” *J. Comput. Phys.*, vol. 25, pp. 220–252, 1997.

[21] W. L. Olbricht, “Pore-scale prototypes of multiphase flow in porous media,” *Annu. Rev. Fluid Mech*, vol. 28, pp. 187–213, 1996.

[22] C. Pozrikidis, “Modeling and simulation of capsules and biological cells,” *Chapman/CRC*, 2003.

[23] M. Muradoglu and S. Gokaltun, “Implicit multigrid computations of buoyant drops through sinusoidal constrictions,” *ASME J. Applied. Mech.*, 2004.

[24] J. H. Ferziger and M. Peric, “Computational methods for fluid dynamics 3rd edition,” *Springer*, pp. 157–264, 2002.

[25] G. Taylor, “Deposition of a viscous fluid on the wall of a tube,” *J. Fluids Mech.*, vol. 10, p. 161, 1961.

[26] F. P. Bretherton, “The motion of long bubbles in tubes,” *J. Fluids Mech.*, vol. 10, p. 166, 1961.

[27] P. Aussillous and D. Quere, “Quick deposition of a fluid on the wall of a tube,” *Physics of Fluids*, vol. 12, pp. 2367–2371, 2000.

[28] M. D. Giavedoni and F. A. Saita, “The axisymmetric and plane cases of a gas phase steadily displacing a newtonian liquid-a simultaneous solution of the governing equations,” *Phys. Fluids*, vol. 9, pp. 2420–2428, 1997.

[29] J. Ratulowski and H. C. Chang, “Transport of gas bubbles in capillaries,” *Phys. Fluids*, vol. 10, pp. 1642–1655, 1989.

[30] D. A. Reinelt and P. G. Saffman, “The penetration of a finger into a viscous fluid in a channel and tube,” *J. Sci. Stat. Comput.*, vol. 6, p. 542, 1985.

[31] M. J. Martinez and K. S. Udell, “Axisymmetric creeping motion of drops through circular tubes,” *J. Fluid Mech*, vol. 210, p. 565, 1990.

VITA

Satayef Kassabbashi was born in Hama, Syria on June 05, 1981 . He graduated from Abi-Elfida High School in 1998. He received the B. Sc. degree in Mechanical Engineering from Middle East Technical University (METU), in 2004. In October 2004, he joined to the Mechanical Engineering Department of Koç University, Istanbul, Turkey as a teaching and research assistant.