1991

Boundary Condition Methods In The Numerical Solution Of The Navier-stokes Equations With The Application To Free Surface Problems

Hua Yang

BOUNDARY CONDITION METHODS
IN THE NUMERICAL SOLUTION OF THE NAVIER-STOKES EQUATIONS
WITH THE APPLICATION TO FREE SURFACE PROBLEMS

By

Hua Yang

Faculty of Engineering Science

Submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario

January, 1991

ISBN 0-315-64278-5

ABSTRACT

This thesis is divided into two main parts. Each is related to the numerical simulation of fluid flows. The first part is concerned with the treatment of pressure or vorticity boundary conditions in the numerical solution of the Navier-Stokes equations; and the second part is concerned with the numerical algorithm for simulating the dynamics of capillary surface.

In Part I, two new accurate finite-difference algorithms for solving the two dimensional, steady, incompressible Navier-Stokes equations are described. The numerical algorithms can be easily adapted on a uniform grid for solving general flows. The first approach, designated as Zero Perturbation Method, uses the combination of the momentum and divergence equations at the boundary to provide the implicit pressure boundary conditions. Its numerical solution procedure is discussed in primitive variable formulation. The second approach, designated as Computational Boundary Condition Method, utilizes a computational solution domain to avoid the problems of no explicit boundary conditions for pressure or vorticity in the Navier-Stokes equations. The pressure or vorticity boundary conditions are implicitly specified on the computational boundaries by the overspecified velocity or stream-function conditions. Both stream function-vorticity and primitive variable formulations are discussed in this case. Although the methods described in this thesis have revealed their wide scope of applications to general flow problems such as three dimensional cases, the primary concern is to provide some general solution procedures that offer reliable, efficient, accurate simulation for all dependent variables in the Navier-Stokes

equations.

Numerical experiments were carried out by second-order finite difference approach and alternating direction implicit (ADI) solution procedure. All tests were performed on uniform grids by means of Zero Perturbation Method and Computational Boundary Condition Method described in this thesis. The computed convergence rates (root mean sqaure) for all variables are in excellent agreement with the theoretical prediction. Results for the classical driven cavity problem are found to be very accurate in comparison to previous investigators. The new methods have not only theoretical significance, but also wide scope of practical applications. In Part II , the analysis has been made on the dynamics of liquids in a low gravity environment which is essentially dominated by its capillary effects. There are two major difficulties in numerical simulation. They are : i) accurate tracking of the curvature of an interface undergoing large deformations, and ii) diagnosing the initiation break-up of an interface, i.e. break-up of a liquid drop. Numerical simulation of these effects has to rely on algorithms capable of handling moving boundary problems for the Navier-Stokes equations. A successful resolution of these difficulties requires an algorithm capable of accurate determination of pressure and velocities along the deformed interface. The required algorithms are described in Part II. This accurate and reliable algorithm for simulation of moving capillary surfaces is given by using Zero Perturbation Method discussed and tested in Part I.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Professor T. E. Base, for his helpful suggestion and guidance during the final stage of this doctoral work. I am particularly indebted to his understanding, as well as kindness of offering the supervision of this research. I also wish to thank Mr. Huaxiong Huang for the fruitful discussion during the final stage of my doctoral work.

This interesting topic was originally related to the research work of Professor S.C.R. Dennis who is a knowledgeable mathematician in Computational Fluid Mechanics. Professor Dennis pointed out the pressure boundary condition problem during my Ph.D. qualified examination. His review paper of integral constraint method in 1989 enlightens me to search for a novel algorithm through a simple one-dimensional model which is the key of my success.

Finally, I wish to thank Professor J. M. Floryan for his guidance and financial support during the first one and half years of my Ph.D. program. I also wish to thank Professor J. D. Tarasuk who offered his kindly understanding and encouragement for this research.

To my mother,

who has been deeply affected by the separation during the period of this study.

TABLE OF CONTENTS

PART I: BOUNDARY CONDITION METHODS IN THE NUMERICAL SOLUTION OF THE NAVIER-STOKES EQUATIONS

CHAPTER 1: INTRODUCTION

1.1 General Review

The numerical simulation of incompressible viscous flow using a discretized system of equations to approximate the Navier-Stokes equations poses one of major difficulties due to the lack of pressure or vorticity boundary conditions. The natural consideration of the discretization of differential equations at the boundaries overdetermines the discretized system. Many methods of determining the appropriate conditions for either primitive variable or stream function-vorticity formulations have been proposed. Unfortunately, there is still a lack of general algorithms favorable to both formulations. In this thesis, two straightforward solution procedures are described which are suitable for all normal discretization of the Navier-Stokes equations.

Finite difference methods for the Navier-Stokes equations can be classified into three major types. These are: the method of staggered grid, the method of projection, and the method of stream function-vorticity. The first approach, known as Mark-and-Cell by Harlow and Welch (1965), uses the staggered grid for imposing the momentum and incompressibility equations. The treatment of the boundary conditions seems easier when this grid arrangement is used, especially, for the pressure. Unfortunately, the lack of pressure boundary condition problems remain unresolved. Instead of solving Poisson equation for pressure, Patankar and Spalding (1977) derived an approximate equation of pressure correction from the momentum equations to satisfy the incompressible equations. The algorithm named as SIMPLE has been widely used in practice.

The second approach is the projection method for the primitive variable

formulation. This method, originally developed by Chorin (1967), sets up an auxiliary vector field V^* in which the momentum and incompressibility conditions are treated in two fractional steps. At the second step, the Neumann condition for pressure is made by projecting the vector field V^* onto its subspace with zero divergence, and satisfying the appropriate boundary conditions. The method makes use of

$$(\nabla p)_\Gamma \cdot \mathbf{n} = \frac{1}{\Delta t}(V^*_\Gamma - V_\Gamma)\cdot \mathbf{n} \qquad \text{(I-1)}$$

as the pressure condition for solving the pressure Poisson's equation. For this method, Alfrink (1981) offered an explicit description of the required compatibility conditions for practical use. In the context of the projection method, Stephens, Bell, Solomon and Hackerman (1985) developed the finite difference Galerkin method for the numerical solutions of steady, incompressible Navier-Stokes equations based on Chorin's original idea. A Galerkin solution procedure involves constructing a local basis for discretized divergence-free vector field, and is used primarily for solutions in terms of primitive variables. Recently, the finite difference Galerkin method has been extended to time dependent incompressible flow. The details may be found in Bell, Colella and Glaz (1990), Goodrich and Soh (1990).

The third approach is to solve the vorticity transport equation by eliminating the pressure from the momentum equations, in which the vorticity is considered as a primary unknown. This stream function-vorticity method has been widely used for calculating the two dimensional flows. An accurate second-order form of implicit vorticity condition was first suggested by Woods (1954). An excellent review for the

treatment of the boundary conditions in this formulation can be found in Roache (1972). In this context, an interesting method that uses the integral type boundary conditions for vorticity was initially given by Quartapelle (1982), and reviewed by Dennis and Quartapelle (1989). The method makes uses of Green's identity to transform locally implicit boundary condition into a global integral constraint. Successful calculations have been made for the flow around the circular cylinder problems as well as many other flow problems.

In recent years, several other interesting methods have been proposed with primitive variable formulation. Among them, the influence matrix method by Kleiser and Schumann (1980) provides a perspective to the problem by introducing a supplementary (linear) problem to determine the missing boundary values for the pressure. This method is quite efficient and has been employed to compute the three dimensional channel flow. The major contribution of Kleiser and Schumann is that their method theoretically ensures the necessary enforcement of the divergence-free constraint at the boundary in solving the pressure Poisson's equation.

While these various methods for the numerical simulation of the Navier-Stokes equations have shown their particular strengths in considering a variety of flow problems, several difficulties still remain in the calculations. Open problems are: i) the staggered grid may result in an inaccuracy in the pressure since both momentum and incompressibility equations are not exactly satisfied at the boundary; ii) the discrete Poisson equation is not strictly equivalent to the incompressibility equation and, therefore, the enforcement of zero-divergence on the boundary may still remain as a dominant issue for the practical use of a primitive variable solution procedure; iii)

because of the complexities of the numerical implementation, some methods are cumbersome for solving problems with complicated geometry, especially, in three dimensions.

1.2 The Navier-Stokes Equations

The equations of motion of steady, viscous, incompressible flow are the Navier-Stokes equations written in a nondimensional form

$$\mathbf{V} \bullet \nabla \mathbf{V} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{V}, \tag{I-2}$$

$$\nabla \bullet \mathbf{V} = 0 \tag{I-3}$$

where $\mathbf{V}$ is the velocity field, p is the pressure and Re is the Reynolds number ($Re = \frac{UL}{\nu}$ where ν is the viscosity). Appropriate boundary conditions include specification of $\mathbf{V}$ at rigid no-slip boundaries or specification of v_n and $\frac{\partial v_\tau}{\partial n}$ at no-stress boundaries when n and τ denote the normal and tangential directions. The pressure field serves to preserve the incompressibility condition (I-3); the divergence of (I-2) gives a Poisson equation for the pressure

$$\nabla^2 p = -\nabla \bullet (\mathbf{V} \bullet \nabla) \mathbf{V}. \tag{I-4}$$

The most widely applicable and popular numerical methods for flow simulation are based on finite difference approximation to (I-2) -(I-4). There are two numerical formulations: one is to solve (I-2)-(I-4) directly in terms of the primitive variables $\mathbf{V}$

and p; another is to solve in term the stream function-vorticity ψ and ζ. In the two space dimensions, the stream function ψ is related to the incompressible velocity V and vorticity ζ by

$$\mathbf{V} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}\right), \qquad \zeta = -\nabla^2 \psi \tag{I-5}$$

while the curl of (I-2) gives

$$\zeta_x \psi_y - \zeta_y \psi_x = \frac{1}{Re} \nabla^2 \zeta. \tag{I-6}$$

The stream function-vorticity formulation is particularly appealing because vorticity is generated locally near boundaries in high Reynolds number flows and subsequently diffused and convected away. On the other hand, the pressure is governed by the elliptic equation (I-4) so that it is affected instantaneously at all points of space. Thus, with limited iterations, it should be easier to reach the convergence state using the stream function-vorticity formulations. However, this expectation is not borne out in practice. Primitive variable formulations tend to be slightly more accurately for similar computational efforts (Orszag 1971) probably because (I-6) requires finite difference approximation to more critical derivatives than does (I-2). Nevertheless, the idea in the stream function-vorticity formulation is still useful for the determination of boundary conditions in primitive variable formulations. The essential part of stream function-vorticity formulation is the inversion of the Poisson equation $\nabla^2 \psi = -\zeta$ for the stream function. It is now recommended that the direct methods such as Fast Fourier

Transformation and Poisson Solver, rather than iterative methods, be used wherever possible. The development of direct methods in the last fifteen years has been one of the big breakthroughs in the developments of efficient simulation codes.

Primitive variable formulations are increasingly widespread, especially in three dimensional simulations. One of convenient methods of using the primitive variable formulation is to replace the incompressible condition (I-3) by the zero-perturbed divergence equation

$$\nabla \bullet \mathbf{V} + \varepsilon p_t = 0 \qquad \text{(I-7)}$$

and to let $\varepsilon \rightarrow 0+$ (Chorin 1967). In this method of artificial compressibility, the fluid is allowed to be slightly compressible and incompressibility is achieved by dynamical relaxation. The technique has proved useful in a variety of flow problems, especially steady state flows (Fortin, Peyret & Temam 1971). It should be pointed out here that the iterative process for the implementation of the algorithm is very straightforward if the treatment of boundary condition is correct. However, with inappropriate boundary conditions, the process may be tricky and lead to difficulties.

An alternative to solving the zero-perturbed divergence equation for the pressure p is to use the Poisson equation (I-4). This approach has been generally considered to be more efficient since it could be conveniently solved by the direct methods referred to earlier. Techniques such as Mark-and Cell Method earlier developed by Harlow & Welch (1965), Influence Matrix Method proposed by Kleiser and Schumann (1980) and Finite Difference Galerkin Method described by Stephen,

Bell, Solomon and Hackerman (1985) have attracted many researchers for their compliance with the incompr- . ibility condition.

1.3 The Treatment of No-Slip Boundary Conditions

It is generally understood that vorticity is generated only at the body surface. For example, a solid body moving in a viscous incompressible fluid under conservative forces generates the vorticity. The generation of vorticity is related to the adherence of the fluid to the solid boundary. This could be expressed mathematically by the no-slip boundary condition $V = V_\Gamma$ at the body surface, where V_Γ is the velocity of the body. The implementation of the no-slip condition is straightforward when primitive variable formulations are u :d as dependent variables. However, the no-slip boundary condition in the stream function-vorticity formulation may require implicit expressions.

We now remark on some uses of the no-slip boundary conditions in primitive variable formulations. The major issue involved in such formulation is the determination of the pressure boundary conditions. In order to solve for pressure p, it is necessary to know p_Γ (the boundary values of the pressure p). First, a Poisson's equation (I-4) is supplemented by prescribed boundary conditions

$$p_\Gamma = B(\zeta) \tag{I-8}$$

where Γ denotes the boundary of the numerical simulation domain and $B(\zeta)$ denotes an unknown pressure distribution o- the boundary. A general solution of (I-4) and (I-8) can be represented as a superposition of an inhomegeneous solution $\tilde{p}$ of (I-4) subject to homogeneous boundary condition and homogeneous solutions of (I-4) subject to the

inhomogeneous boundary condition (I-8). Thus the pressure can be written as

$$p = \tilde{p} + \sum_{i}^{M} \lambda_i p_i \tag{I-9}$$

where i denotes a discretization point along the boundary and the inhomogeneous boundary conditions have been selected in the form (i.e., ζ_i are the discrete points at the boundary)

$$p_\Gamma|_{\zeta = \zeta_i} = 1, \qquad p_\Gamma|_{\zeta \neq \zeta_i} = 0. \tag{I-10}$$

In the above, λ_i are the constants of superposition to be determined from the incompressibility condition at the boundaries. It can be shown through the application of Green's theorem that this assures fulfillment of incompressibility condition in the whole flow field. Equations for determination of λ_i, the influence matrix (Kleiser and Schumann 1980), are constructed by representing velocity vector as

$$\mathbf{V} = \tilde{\mathbf{V}} + \frac{1}{Re} \sum_{i}^{M} \lambda_i \mathbf{V}_i \tag{I-11}$$

where $\tilde{\mathbf{V}}$ is a solution of a Stokes problem with inhomogeneous no-slip boundary condition $\mathbf{V} = \mathbf{V}_\Gamma$ and $\mathbf{V}_i$ are the solutions of the Navier-Stokes equation written in Stokes form with homogeneous no-slip boundary conditions $\mathbf{V} = 0$. Eq. (I-11) is used to construct the divergence operator and enforcement of the zero-divergence condition at grid points along the boundaries gives the required equations for λ_i.

If the steady Navier-Stokes equations (I-2) are written in unsteady forms as $\mathbf{F} = \mathbf{w} + \nabla p$, $\mathbf{w} = \frac{\partial \mathbf{v}}{\partial t}$ where $\mathbf{F}$ denotes the convective, diffusive and external-force

terms, then $\mathbf{w}$ and ∇p are the solenoidal components, respectively, in the Weyl decomposition of $\mathbf{F}$ (Ladyzhenskaya 1969). Since $\mathbf{w}$ and ∇p are determined uniquely by $\mathbf{F}$ and specification of either the normal component w_n or the tangential component w_τ along the boundary, it follows that exact solutions to the Navier-Stokes equations must satisfy an auxiliary dynamical consistency condition to permit specification of both w_n and w_τ. This is similar to the idea of Chorin (1968) that the momentum equations can be split into a divergence free and rotational free part. If the explicit finite differences are used, this yields the following scheme

$$\frac{\mathbf{V}^* - \mathbf{V}^n}{\Delta t} + \mathbf{V}^n \bullet \nabla \mathbf{V}^n = \frac{1}{Re}\nabla^2 \mathbf{V}^n, \tag{I-12}$$

$$\nabla^2 p = \frac{\nabla \bullet \mathbf{V}^*}{\Delta t}, \tag{I-13}$$

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^*}{\Delta t} + \nabla p^{n+1} = 0. \tag{I-14}$$

First of all, the momentum equations excluding the pressure gradient are solved, yielding an auxiliary velocity field, which contains the correct vorticity but has non-zero divergence. The auxiliary velocity is then modified in such a way as to bring the divergence to zero while preserving the vorticity. Using Eq.(I-14) the boundary condition for pressure can be written here as

$$(\nabla p)_\Gamma \bullet \mathbf{n} = \frac{1}{\Delta t}(\mathbf{V}^*_\Gamma - \mathbf{V}_\Gamma) \bullet \mathbf{n}. \tag{I-15}$$

The scheme proposed has been used amongst others by Fortin *et al* (1971). The right hand side of Eqs.(I-13) and (I-15) must satisfy the following compatibility condition, obtained by applying Gauss' theorem

$$\int_{\Omega} \frac{\nabla \bullet \mathbf{V}^*}{\Delta t} d\Omega = \oint_{\Gamma} \frac{\mathbf{V}^* - \mathbf{V}^{n+1}}{\Delta t} \bullet \mathbf{n} d\Gamma, \tag{I-16}$$

where Ω denotes the two dimensional domain under the consideration with its boundary Γ. In case there is conservation of mass, i.e., $\oint_{\Gamma} \mathbf{V}^{n+1} \bullet \mathbf{n} d\Gamma = 0$, Eq.(I-16) reduces to

$$\int_{\Omega} (\nabla \bullet \mathbf{V}^*) \, d\Omega = \oint_{\Gamma} \mathbf{V}^* \bullet \mathbf{n} d\Gamma. \tag{I-17}$$

The specific example is given by Kim and Moin (1985). In the context of Galerkin ideas, Stephens, Bell, Solomon and Hackerman (1984) use their finite difference Galerkin method for the system (I-12)-(I-14). They begin with a discretization of equations of motion, and then they manipulate the finite difference equations and their solution space to obtain a convenient algorithm. The essential features of their method are the expansion of the discrete velocity solution using a basis for the discretely divergence free vector fields on the grid and derivation of equations for the expansion coefficients by taking the inner product of the expansion vectors and the discrete momentum equations. The only exceptional constraint is that the primitive variable discretization for the divergence operator in the incompressibility equation must be the adjoint (matrix transpose) of the discretization for the gradient operator applied to the pressure in the momentum equations. If the discrete divergence and gradient operators are the adjoints of each other, then the discrete pressure will drop out of the derived equations. The essential problems in applying such method are to find a basis for the null-space of the discrete divergence operator and to find a particular solution of the discrete incompressibility equation that accounts for the velocity boundary values that

are prescribed by the problem that is being solved.

The relevant ideas of the earlier work in the stream function-vorticity formulation are easily introduced by a steady Stokes flow. In this case, the substitution of a stream function gives the biharmonic equation

$$\nabla^4 \psi = 0. \tag{I-18}$$

On a stationary rigid wall, the no-slip conditions are

$$\psi = 0, \quad \frac{\partial \psi}{\partial n} = 0 \tag{I-19}$$

where n is the coordinate normal to the boundary. Pearson (1965) showed that it is computationally more efficient to split the fourth order equation into the coupled system of Laplacian and Poisson equations

$$\nabla^2 \zeta = 0, \qquad \zeta = -\nabla^2 \psi. \tag{I-20}$$

In order to solve for ζ from $\nabla^2 \zeta = 0$, it is necessary to know ζ_Γ (the boundary values of the vorticity ζ). Various difference approximations for ζ_Γ in terms of the vorticity and stream function in the interior of the region have been suggested; when the Lapalacians are also discretized a closed system linear equations for grid point values results, which may be solved by relaxation. This procedure was used by Thom (1933), Payne (1958), Jenson (1959), and many others.

When a time-dependent solution procedure is considered, the proper analogue

of the biharmonic equation is

$$\frac{\partial \zeta}{\partial t} = \frac{1}{Re}\nabla^2 \zeta, \qquad \zeta = -\nabla^2 \psi. \tag{I-21}$$

Two approaches are possible now: explicit or implicit. In the explicit scheme, ζ^{n+1} (the vorticity at the n+1 step) is computed first at all the interior grid points from the equation

$$\zeta^{n+1} - \zeta^n = \frac{\Delta t}{Re}\nabla^2 \zeta^n, \tag{I-22}$$

The elliptic equation $\zeta^{n+1} = \nabla^2 \psi^{n+1}$ is solved for ψ^{n+1} and then ζ_Γ is updated. In the implicit method, one can use the Crank-Nicolson scheme

$$\zeta^{n+1} - \frac{1}{2Re}\Delta t \zeta^{n+1} = \zeta^n + \frac{1}{2Re}\Delta t \nabla^2 \zeta^n, \tag{I-23}$$

$$\zeta_\Gamma^{n+1} = L(\psi^{n+1}, \zeta^{n+1}), \tag{I-24}$$

where L is a linear combination of its arguments, or the alternating-direction-implicit (ADI) method due to Peaceman and Rachford. In both cases an iterative process is required. If Cartesian coordinates (x,y) are used and the boundary is the y axis, it follows that $\zeta_\Gamma = \left(\nabla^2 \psi\right)_\Gamma = \left(\frac{\partial^2 \psi}{\partial x^2}\right)_\Gamma$ because $\psi_\Gamma = 0$ implies that $\left(\frac{\partial^2 \psi}{\partial y^2}\right)_\Gamma = 0$. A second-order central difference approximation gives

$$\zeta_\Gamma = \left(\frac{1}{\Delta x^2}\right)(\psi(\Delta x) - 2\psi(0) + \psi(-\Delta x)), \tag{I-25}$$

where $\Delta x = -\Delta x$ is outside the region of computation but $\psi(-\Delta x)$ can be computed

using the condition $(\frac{\partial \psi}{\partial x})_{\Gamma} = 0$.

Thus Fromm (1963) uses

$$0 = (\frac{\partial \psi}{\partial x})_{\Gamma} = (\frac{1}{\Delta x})\,(\psi(0) - \psi(-\Delta x)) + O(\Delta x)\,, \tag{I-26}$$

so that since $\psi(0) = 0$

$$\zeta_{\Gamma} = \frac{\psi(\Delta x)}{\Delta x^2} + O(1)\,. \tag{I-27}$$

Thom (1933) uses $\psi(-\Delta x) = \psi(\Delta x) + O(\Delta x^3)$, which gives

$$\zeta_{\Gamma} = \frac{2\psi(\Delta x)}{\Delta x^2} + O(\Delta x)\,. \tag{I-28}$$

Woods (1954) obtained the equivalent of

$$\zeta_{\Gamma} = \frac{3\psi(\Delta x)}{\Delta x^2} - \frac{1}{2}\zeta(\Delta x) + O(\Delta x^2)\,. \tag{I-29}$$

The truncation errors in the expressions were obtained by formal manipulation of the Taylor series (Roache 1972). Similar ideas were employed by Ghia *et al* (1982) who used the third-order form of implicit condition in their efficient multigrid solution procedure.

Another interesting treatment of the no-slip boundary condition is the idea of the integral constraint which uses the integral type boundary conditions for vorticity (Dennis and Quartapelle 1989). It makes uses of Green's identity to transform locally

implicit boundary condition into a global integral constraint. To illustrate the concept, a model problem is constructed to analogue the problems of the Navier-Stokes type. Consider the functions $\psi(x)$ and $\zeta(x)$ which satisfy the one-dimensional system

$$\zeta'' + f(x)\,\zeta' = g(x) \tag{I-30}$$

$$\psi'' = \zeta, \tag{I-31}$$

with boundary conditions:

$$\psi = \psi' = 0 \qquad \text{when } x = 0, 1. \tag{I-32}$$

Multiplying (I-31) by its fundamental solutions ($\psi = 1$ and $\psi = x$) in turn and integrating with respect to x from $x = 0$ to $x = 1$, one obtains the two global conditions for the vorticity,

$$\int_0^1 \zeta dx = 0\,; \qquad \int_0^1 x\zeta dx = 0. \tag{I-33}$$

Equations (I-33) provides the relationship between the boundary vorticity ζ_Γ and interior values of the vorticity. However, the numerical implementation of such idea in two-dimensional cases could lead to some difficulties (Dennis and Quartapelle 1989).

1.4 Finite difference, Finite element and Spectral method

There are three particularly appealing methods in the modern development of numerical simulations of the Navier-Stokes equations. These are: finite difference, finite element and spectral method. Finite element method uses local, low-order

polynomial basis functions to generate sparse algebraic equations in terms of nodal unknowns; spectral method makes use of global, orthogonal basis functions to achieve a high accuracy per degree of freedom; finite difference method, however, offers a more direct approach to the numerical solution of the partial differential equations. In the latter approach, one simply replaces the derivatives with finite difference expansion and requires that the resulting algebraic equations be satisfied exactly at the grid points. The finite difference method can be interpreted as a collocation method without a basis solution. As we have discussed earlier, the difficulties in the finite difference method arise in imposing boundary conditions, and low-order finite difference formulations are often inaccurate and computationally expansive, particularly on coarse grids.

One of the major concerns in these methods is the computational efficiency. A number of different ways of defining computational efficiency are discussed by Swartz (1974). In the aspect of the influence of accuracy and economy on the computational efficiency, the following simple definition may be used:

$$\text{Computational Efficiency} = \frac{k}{\varepsilon\tau}$$

where ε is the error in the computed solution in some appropriate norm, and τ is the execution (CPU) time or operation count. Thus a solution of coarse grid that has a large error but a small execution time is as computationally efficient as a solution on refined grid that produce a small error but requires a large execution time. Since computationally efficiency is a property of the method, the relative powers of ε and τ should be chosen so that the computational efficiency is independent of grid refinement.

The three categories of computational methods: finite difference, finite element and spectral methods can be ranked in that order as far as economy per degree of

freedom is concerned. This would be expected from the relative sparseness of algebraic equations formed by methods if the same total number of degrees of freedom are involved. From a consideration of the error estimates it would be expected that the accuracy achieved by the various methods with the same number of degrees of freedom could be ranked as follows: spectral, finite element and finite difference methods. However, the relative computational efficiency of the various methods is not obvious from fundamental considerations. The efficiency will depend on the particular problem being considered, the order of the basis solution, the number of dimensions, and the influence of the aspects of the computational algorithm. It is generally recognized that for higher-order, higher-dimensional local methods the algebraic complexity is much greater for the finite element methods than the finite differences. Consequently it may be expected the economy to be significantly worse and require the accuracy to be significantly higher if the computational efficiency is to be greater.

Now we make some remarks on simplicity of the programming codes and flexibility. From the nature of finite difference method we would expect it to require relatively little algebraic manipulation and relatively straightforward programming codes, except that special procedures might be required for particular boundary conditions. Generally, finite difference method is inflexible, since we would expect very little of the existing program to be relevant to the next problem. However, the relative simplicity of the programming applies to the second problem just as much as the first. Finite element method requires some basic algebraic manipulation and more programming effort than finite difference method. However, the modularity of the finite element method leads itself to efficient programming, and a lot of finite element packages exist. Flexibility is an important feature of the finite element method. In solving a new problem relatively few changes need to be made in an existing package.

Spectral method requires considerable amount of basic algebraic manipulation and programming if an efficient code is to be generated. Also, the solution of a new problem typically requires a new basis solution, new specification of boundary conditions. Therefore, spectral method is relatively difficult to program and inflexible.

1.5 Outline of Present Work

This thesis is divided into two main parts. Each is related to the numerical simulation of fluid flows. The first part is concerned with the treatment of pressure or vorticity boundary conditions in the numerical solution of the Navier-Stokes equations; and the second part is concerned with the numerical algorithm for simulating the dynamics of a capillary surface.

It is of interest that when the doctoral research was initiated the study was concerned with a two dimensional free/moving surface problem in a cavity which is described in Part II. However, after considering the existing numerical algorithm for this problem, it was realized that the major difficulties in accurately simulating the free surface flow are: i) the correct treatment of the pressure boundary condition, and ii) the use of the uniform discretization scheme to solve the Navier-Stokes equations in primitive variable formulation. There is no appropriate algorithm suitable for solving such kind of flows. This leads to an in depth study of boundary condition methods in the numerical solution of the Navier-Stokes equations since there is no explicit pressure or vorticity boundary condition. This analysis is discussed in Part I which has become a major part of this doctoral work.

Outline of Part I

In this part two new accurate finite-difference algorithms for the two dimensional,

steady, incompressible Navier-Stokes equations are described. The numerical algorithms can be easily adapted on uniform grid for solving general flows. The first approach, designated as Zero Perturbation Method, uses the combination of the momentum and divergence equations at the boundary to provide the implicit pressure boundary conditions. The numerical solution procedure is discussed in primitive variable formulation. The second approach, designated as Computational Boundary Condition method, utilizes a computational solution domain to avoid the problems of no explicit boundary conditions for pressure or vorticity in the Navier-Stokes equations. The pressure or vorticity boundary conditions are implicitly specified on the computational boundaries by the overspecified velocity or stream-function conditions. The latter approach seems more natural for the general flow problems. In this case, both stream function-vorticity and primitive variable formulations are discussed. In both finite-difference approaches, the overdetermined system resulting from the Navier-Stokes equations can be correctly reduced to a determined system. The methods in this thesis have revealed their wide scope of applications to general flow problems such as three dimensional cases. However, the general concern in this part is only to provide some general solution procedures that offer reliable, efficient, accurate simulation for all dependent variables in the Navier-Stokes equations.

Due to lack of the pressure or vorticity boundary conditions in the simulation, the common solution procedure of discretizing equation at the boundary results in an identity (or algebraic dependence) in the discretized equations. Therefore, the correct boundary conditions for such a system is difficult to be found. This boundary value problem in the Navier-Stokes equations has been studied by many researchers for more than twenty years. The theory and numerical illustration of such an identity (or algebraic dependence) will be given and discussed later in chapter two and chapter four

respectively. This has indicated that the major difficulties involved in solving flow problems for both primitive variable and stream function-vorticity equations are essentially similar. The difficulties can be easily removed by using the new approaches proposed in this thesis. The proposed methods are designated as Zero Perturbation Method and Computational Boundary Condition Method. The implementation of such ideas is convenient and straightforward for the Navier-Stokes equations. The focus on the first part of the thesis is to provide a correct determined system of equations that all the difference equations could be imposed. In chapter two, the proof of equivalence between the system of the Navier-Stokes equations and a modified Navier-Stokes system is presented. A detailed description of its finite difference implementation is also included. Chapter three presents the one dimensional illustrations for both stream function-vorticity and primitive variable formulations for the Computational Boundary Condition Method and then, the applications to general two dimensional flow problems are followed. The numerical results are given in chapter four which carries out several numerical experiments for both one and two dimensional model problems to show the accuracy and capabilities of our new methods. The summary is given in chapter five.

CHAPTER 2: ZERO PERTURBED BOUNDARY CONDITION METHOD

2.1 Second-Order Extrapolation

Let h be the uniform grid size of the discretization scheme in a one-dimensional interval (a,b) and its boundaries a and b; and let

$$\{\ x_i = a + (i\text{-}1)h,\quad 1 \le i \le M+1,\ \textit{where}\ M = (b-a)/h\ \} \tag{I-34}$$

be the discrete points in $[a,b]$.

We define the function $f(x)$ that $f^{(2)}(x)$ is continuous in $[a, b]$, and $f^{(3)}(x)$ exists in (a,b) where { x defined on $[a, b]$ } $\supseteq$ { x_i defined by (I-34) }. Their values at discrete points x_i are written as $f_i = f(x_i)$. We notice that $f_1 = f(a)$ represents the value at the boundary a where $a = x_1$.

Suppose the values $f_i = O(h^2)$ ($i = 2, 3$) are given; Suppose $f_1 = O(h^{\alpha})$ is given. Then i) the extrapolation to f_1 given by f_2, f_3 written as

$$f_1 = 2f_2 - f_3 + Ch^2 \tag{I-35}$$

is a second-order approximation, where C is a constant which is independent of f_i (i = 1, 2, 3);

and ii) $f_1 = O(h^2)$.

Since f_i (i = 1, 2, 3) represent the values of $f(x)$ at points x_i (i = 1, 2, 3), one may construct a parabolic interpolation by f_i (i = 1, 2, 3) approaching to $f(x)$ in $[x_1, x_3]$ written as

$$P_2(x) = \sum_{k=1}^{3} \left(\prod_{\substack{i=1 \\ i \neq k}}^{3} \frac{(x - x_i)}{(x_k - x_i)} \right) \cdot f_k , \tag{I-36}$$

and its error is determined by

$$R_2(x) = f(x) - P_2(x) = \frac{f^{(3)}(\xi)}{3!} \cdot \prod_{k=1}^{3} (x - x_k) = O(h^3), \tag{I-37}$$

where $x_1 \leq \xi \leq x_3$. It is obvious that in the limit of h $\rightarrow$ 0: i) $P_2(x)$ tends to $f(x)$ and ii) $P_2(x)$ exists.

$f^{(2)}(x)$ is continuous in $[a, b]$ and $f^{(3)}(x)$ exists in (a,b). Therefore, from (I-36) and (I-37) one may conclude that i) $P_2^{(2)}(x)$ is continuous in $[x_1, x_3]$ and ii) $P_2^{(2)}(x)$ exists in (x_1, x_3).

By taking the second-order differentiations of Eq. (I-36) with respect to x, it becomes

$$P_2^{(2)}(x) = \frac{f_1 - 2f_2 + f_3}{h^2}, \qquad \text{in } (x_1, x_3). \tag{I-38}$$

Substituting the conditions $f_1 = O(h^{\alpha})$ and $f_i = O(h^2)$ (i = 2, 3) into (I-38), one may obtain

$$P_2^{(2)}(x) = \frac{f_1 - 2f_2 + f_3}{h^2} = O(h^{\alpha-2}), \qquad \text{in } (x_1, x_3). \tag{I-39}$$

In the limit of h → 0, Eq. (I-38) states that $P_2^{(2)}(x)$ exists and therefore, the possible value of $P_2^{(2)}(x)$ in Eq. (I-38) is

$$C(f_i) = \frac{f_1 - 2f_2 + f_3}{h^2}, \tag{I-40}$$

where the value of $C(f_i)$ depends on the value of f_i (i = 1, 2, 3).

Eq. (I-39) implies that i) $P_2^{(2)}(x)$ is given by f_i (i = 1, 2, 3) and ii) α must be greater than or equal to 2 (otherwise $P_2^{(2)}(x)$ will not exist). In other words, the constant C is independent of f_i and exists in (x_1, x_3). Thus solving Eq. (I-40) for f_1 one may obtain Eq. (I-35).

We should point out that the conventional second-order extrapolation gives an identical formula $f_1 = 2f_2 - f_3 + C(f_1)h^2$ which only provides an indefinite value of C = C(f_1) in the equation. Therefore, f_1 may not be assured as $O(h^2)$ since the solutions are indefinite. However, with the constraint $f_1 = O(h^\alpha)$, this ensures that $f_1 = O(h^2)$.

2.2 The Equivalence of Two Algebraic Systems of Equations

The steady, two-dimensional Navier-Stokes equations for an incompressible viscous flow in a domain Ω and its boundary Γ are written as

$$\mathbf{V} \bullet \nabla \mathbf{V} = -\nabla \mathbf{p} + Re^{-1} \nabla^2 \mathbf{V}, \tag{I-41}$$

$$\nabla \bullet \mathbf{V} = 0. \tag{I-42}$$

The velocity $\mathbf{V} = (u,v)$ on the boundary Γ are usually given in forms of Dirichlet boundary conditions associated with the Navier-Stokes equations. Namely

$$\mathbf{V} = \mathbf{V}_{\Gamma}. \tag{I-43}$$

We define this set of governing equations which describes the general fluid motion as problem A.

Problem A is equivalent to the following set of governing equations

$$\mathbf{V} \bullet \nabla \mathbf{V} = -\nabla \mathbf{p} + Re^{-1} \nabla^2 \mathbf{V}, \qquad \text{in } \Omega \tag{I-44}$$

$$\nabla \bullet \mathbf{V} = 0; \qquad \text{in } \Omega \tag{I-45}$$

and

$$\mathbf{V} = \mathbf{V}_{\Gamma}, \qquad \text{on } \Gamma \tag{I-46}$$

$$\begin{aligned} \nabla \bullet \mathbf{V} + w_1 \left(u u_x + v u_y + p_x - Re^{-1} \left(u_{xx} + u_{yy} \right) \right) , \\ + w_2 \left(u v_x + v v_y + p_y - Re^{-1} \left(v_{xx} + v_{yy} \right) \right) = 0 ; \end{aligned} \qquad \text{on } \Gamma \tag{I-47}$$

provided that the velocity $\mathbf{V}$, *pressure* p *and their derivatives are continuous in domain* Ω *and its boundary* Γ, *where* w_1, w_2 *are arbitrary constants.*

As a matter of fact, the system of equations (I-44)-(I-47) is the modified form of problem A. For convenience, we define this modified system (I-44)-(I-47) as Problem B. They are similar except the additional zero-perturbed type of boundary conditions imposed at the boundary.

If V and p are the solutions of problem A, V and p satisfy Eqs. (I-44)-(I-47). Conversely, all the terms in momentum and incompressibility equations are continuous since V, p and their derivatives are continuous in domain Ω and on boundary Γ. If V and p satisfy problem B, the limits of Eqs. (I-44)-(I-45) to its boundary, which are the momentum and incompressibility equations on the boundary, will hold on its boundary Γ. Therefore, V and p are also the solutions of problem A. The boundary condition (I-47) of problem B furnishes the constraint of pressure derivatives on the boundary which provides a unique way to overcome the difficulty of the lack of pressure conditions in the numerical simulation of the Navier-Stokes equations, however, the equivalence of problem A and problem B does not lead to a rigorous equivalence of their discretized forms. We give the following discussion for the equivalence between the two discrete systems.

Let h be the uniform grid size of the discretization scheme in x-, y- direction; and let

$$\Omega_h \cup \Gamma_h = \{ x_i=(i-1)h,\ y_j=(j-1)h,\ 1 \le i \le M+1,\ 1 \le j \le N+1 \} \qquad \text{(I-48)}$$

in a rectangular domain; then Eqs. (I-41)-(I-42) approximated by the discretized system of equations with second-order accuracy could be expressed as

$$u_{i,j} D_x u_{i,j} + v_{i,j} D_y u_{i,j} + D_x p_{i,j} - Re^{-1} (D_{xx} u_{i,j} + D_{yy} u_{i,j}) = 0, \qquad \text{(I-49)}$$

$$u_{i,j} D_x v_{i,j} + v_{i,j} D_y v_{i,j} + D_y p_{i,j} - Re^{-1} (D_{xx} v_{i,j} + D_{yy} v_{i,j}) = 0, \qquad \text{(I-50)}$$

$$(D_x u_{i,j} + D_y v_{i,j}) = 0, \tag{I-51}$$

which satisfy domain Ω_h and its boundary Γ_h. In the above, D_x, D_{xx}, D_y and D_{yy} represent the difference operators. Eqs. (I-49)-(I-51) with the velocity condition (I-43) are defined as Problem A*.

We shall use $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ to represent the residuals (or the truncation error variations) of x-, y- momentum and incompressibility equations respectively, which are defined as the difference between the equations of actual solutions and the numerical solutions. Here, the subscript h denotes the size of uniform mesh, x_i, y_j are the discretized space coordinates. It is understandable that $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ are the functions of grid size h, the discretized space coordinates x_i, y_j and their values also depend on the specification of discretization scheme.

Problem A is equivalent to the following set of discretized system of equations defined as problem B*:*

discretized equations in domain Ω_h

$$Ru_h = u_{i,j}D_x u_{i,j} + v_{i,j}D_y u_{i,j} + D_x p_{i,j} - Re^{-1}(D_{xx}u_{i,j} + D_{yy}u_{i,j}) = 0, \tag{I-52}$$

$$Rv_h = u_{i,j}D_x v_{i,j} + v_{i,j}D_y v_{i,j} + D_y p_{i,j} - Re^{-1}(D_{xx}v_{i,j} + D_{yy}v_{i,j}) = 0, \tag{I-53}$$

$$D_h = D_x u_{i,j} + D_y v_{i,j} = 0; \tag{I-54}$$

discretized equations on boundary Γ_h

$$D_h + w_1 (Ru_h) + w_2 (Rv_h) = 0, \tag{I-55}$$

with the velocity condition (I-46), where w_1, w_2 *are arbitrary constants with order of* $O(1)$;

if i) the discrete residual functions $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ *are well-behaved functions near the boundary* Γ_h,

ii) the difference equations in Problem B are stable.*

When the difference equations of Problem B* are convergent to Problem B, the momentum and incompressibility equations at the boundary may not be necessarily convergent as second-order accuracy. In other words, our uncertainty is whether or not the higher values of cancellations occur among the residual functions $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ in the numerical calculation of Eq. (I-55) at the boundaries. However, this theorem eliminates these possible exceptions which assures the equal rate of convergence for both field and boundary.

First of all, the system of difference equations (I-52)-(I-55) in addition with (I-46) is a closed system, in which, the number of difference equations is the same as the number of the unknowns. For the rectangular domain $\Omega_h \cup \Gamma_h$ defined by (I-48), the difference scheme with uniform grid size in x- and y- directions gives (M+1)×(N+1) points to determine the 3×(M+1)×(N+1) unknowns. Applying the momentum and

incompressibility equations (I-52)-(I-54) in domain D_h at the grid points yields 3×(M-1)×(N-1) equations, leaving 6×(M+N) equations to close the system. The velocity condition (I-46) at Γ_h provides 4×(M+N) equations. The additional 2×(M+N) equations are supplied by Eq. (I-55). This leads to a determined system and therefore, ***problem B**** is a closed system. Secondly, in the limit of $h \to 0$, the difference equations (I-52)-(I-54) represent Navier-Stokes equations. Thus when the convergence state for ***problem B**** is reached, the enforcement of equations (I-52)-(I-54) ensures that: i) all residuals $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ in domain Ω_h are accuracies of $O(h^2)$; ii) all the unknowns in the difference system are determined since it is a closed system. Therefore the residuals $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ at the boundaries are definite values. For example, one may write the order of x-momentum residual $Ru_h(x_i, y_j)$ as $O(h^\alpha)$. Thirdly, the residual is defined as the difference between the equation of actual solution and the numerical solution. It is understandable that it is a (n)th order differentiable function since the (n)th order polynomials are utilized in the finite difference approach, and also, the actual solution is usually assumed as a smooth function. Let $f = f(x,y)$ be a continuous function in domain Ω where $\{(x,y)$ defined on $\Omega \cup \Gamma\} \supseteq \{(x_i, y_j)$ defined by (I-48)$\}$; let $f(x_i,y_j)$ stand for residual function $Ru_h(x_i, y_j)$. Thus in the second-order difference approach, $\frac{\partial^2}{\partial x^2} f(x, y)$ and $\frac{\partial^2}{\partial y^2} f(x, y)$ exist. We define f_1 as the value of residual $Ru_h(x_i, y_j)$ at the boundary Γ_h; f_2 and f_3 as the residuals in domain Ω_h which are close enough to the boundary point f_1 such that the Taylor expansion is valid. We may assume that $f_1 = O(h^\alpha)$ where its value is given by the determined system as mentioned in the above and also, $f_2 = O(h^2)$ and $f_3 = O(h^2)$ because of the uses of the second order

difference approximation. From the conclusion of Section 2.1, one may conclude that α must be greater than or equal to 2. Thus the residual $Ru_h(x_i, y_j)$ at the boundary Γ_h is the order of $O(h^2)$. For example, the extrapolation to the boundary value $Ru_h(x_1, y_j)$ could be approximated by the interior residuals $Ru_h(x_2, y_j)$ and $Ru_h(x_3, y_j)$ as follows

$$Ru_h(x_1, y_j) = 2Ru_h(x_2, y_j) - Ru_h(x_3, y_j) + \frac{\partial^2}{\partial x^2} Ru_h(x_1, y_j)\, h^2; \tag{I-56}$$

and

$$\left| \frac{\partial^2}{\partial x^2} Ru_h(x_1, y_j) \right| < A_h \tag{I-57}$$

where $\frac{\partial^2}{\partial x^2} Ru_h(x_1, y_j)$ denotes the second-order partial differentiation of $Ru_h(x_1, y_j)$ with respect to x. The constant A_h depends on grid size h, and its value could be given by the discretized system. Condition i) states that the residual functions are well-behaved near the boundaries which indicate their derivatives are small enough. This ensures that the constant A_h is definite. Therefore,

$$Ru_h(x_2, y_j) = O(h^2),\ Ru_h(x_3, y_j) = O(h^2),\ \Rightarrow Ru_h(x_1, y_j) = O(h^2); \tag{I-58}$$

similarly, we have

$$Rv_h(x_2, y_j) = O(h^2),\ Rv_h(x_3, y_j) = O(h^2),\ \Rightarrow Rv_h(x_1, y_j) = O(h^2); \tag{I-59}$$

$$D_h(x_2, y_j) = O(h^2),\ D_h(x_3, y_j) = O(h^2),\ \Rightarrow D_h(x_1, y_j) = O(h^2); \tag{I-60}$$

which give second-order approximations to momentum and incompressibility equations at the boundaries. Similar proofs could be made for the residuals at other boundaries. So the residuals $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ at the boundaries are the orders of $O(h^2)$ respectively. Finally, the difference equations in Problem B* may NOT be stable if there are any higher values of cancellations than $O(h^2)$ among the residual functions $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ in Eq.(I-55) at the boundary Γ_h. In other words, when the convergent state of Problem B* is reached, then either all the residuals in Eq.(I-55) are the order of $O(h^2)$ or the difference equations are numerically unstable. Suppose the subscript (1,j) (j=2,..,N) denote the boundary points. If there are any higher values of cancellations, i.e., in

$$w_1 Ru_h(x_1, y_j) + w_2 Rv_h(x_1, y_j) + D_h(x_1, y_j) = O(h^2) , \tag{I-61}$$

then their neighbor points (1,j-1) or (1,j+1) exist the higher value terms than $O(h^2)$ since

$$Ru_h(x_1, y_j) = \frac{Ru_h(x_1, y_{j+1}) + Ru_h(x_1, y_{j-1})}{2} + O(h^2) , \tag{I-62}$$

$$Rv_h(x_1, y_j) = \frac{Rv_h(x_1, y_{j+1}) + Rv_h(x_1, y_{j-1})}{2} + O(h^2) , \tag{I-63}$$

$$D_h(x_1, y_j) = \frac{D_h(x_1, y_{j+1}) + D_h(x_1, y_{j-1})}{2} + O(h^2) . \tag{I-64}$$

Thus, this higher value of cancellation leads to another cancellation among one of its neighbors since its residual functions are also restrained by Eqs. (I-55). Therefore, this may affect the stability of the difference equations in Problem B*.

The equivalence of the two system relies on the appropriate discretization scheme such that consistency and stability conditions for Problem B* are satisfied. The stability conditions are related to the basis that the discrete residual functions $Ru_h(x_i, y_j)$, $Rv_h(x_i, y_j)$ and $D_h(x_i, y_j)$ are well-behaved, especially, near the boundary. Normally, it is difficult to have a strict definition of a well-behaved function. Therefore, our assessments of stability are basically on the reliance of the numerical evidence, i.e., the improper design of the discretization scheme may result in the possible instability of difference equations in which the amplification of the round-off errors may occur. A consistent discretization scheme near the boundary is preferred because this type of approximations may generate relatively smooth values of residual that assure the residual functions are well-behaved near the boundary. In regard to the consistency of the discretization scheme, numerically we may use the following way to evaluate the effectiveness of existing algorithms. For example, we may perform the analysis of global convergence on the results to demonstrate the rate of its convergence. In practice, the numerical tests have indicated that by using this finite difference approach, the computed results are obtained within the expected accuracy. In other words, the stability is not an essential problem in this approach.

2.3 Zero Perturbation Method and Its Numerical Implementation

We write the Navier-Stokes equations and the incompressibility equation in forms of the artificial compressibility method in domain Ω and its boundary Γ where c is an arbitrary constant ($1/c^2 << 1$)

$$u_t + uu_x + vu_y + p_x - Re^{-1}(u_{xx} + u_{yy}) = 0, \tag{I-65}$$

$$v_t + uv_x + vv_y + p_y - Re^{-1}(v_{xx} + v_{yy}) = 0, \tag{I-66}$$

and

$$p_t + c^2(u_x + v_y) = 0. \tag{I-67}$$

The principle of the method is to consider the steady solution as the limit when $t \rightarrow \infty$ of the solution of unsteady equations associated with a perturbed incompressibility equation. Temam (1968) proved that for $1/c^2$ fixed, Eqs. (I-65)-(I-67) have an unique solution which tends to the solution of Navier-Stokes equations. If $1/c^2$ is small enough, the approximate solution would be made.

By constructing the linear combination of momentum and incompressibility equations at the boundary Γ as pressure conditions given in Section 2.2, we replace Eq. (I-67) by the following equation:

$$p_t + c^2(D + w_1 Ru + w_2 Rv) = 0, \qquad \text{on } \Gamma. \tag{I-68}$$

Here, D, Ru, and Rv represent the incompressibility, x- and y- momentum residuals at

the boundary respectively; w_1 and w_2 are chosen as either 1 or -1 to ensure the difference equations stable. We designate the present method as "zero perturbation method" for solving Eqs.(I-65)-(I-67) in domain Ω and Eq. (I-68) on boundary Γ. According to the theory discussed earlier, the numerical solution by zero perturbed momentum and divergence boundary conditions will be the same as the numerical solution of Navier-Stokes equations.

The system solved by zero perturbation method can be used with various difference scheme. In the present case, we solve Eqs. (I-65)-(I-67) in domain Ω by standard ADI solution procedure using second-order central difference for all the derivatives except higher-order upwind scheme (Fletcher 1988) (based on four-point difference scheme) for the convective terms, i.e., the central difference formulae written as,

$$D_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} \tag{I-69}$$

$$D_{xx} u_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \tag{I-70}$$

and the high-order upwind schemes,

$$u_{i,j} D_x u_{i,j} = u_{i,j} \frac{u_{i+1,j} - u_{i-1,j}}{2h} + q u_{i,j} \frac{u_{i-2,j} - 3u_{i-1,j} + 3u_{i,j} - u_{i+1,j}}{3h}, \quad \text{if } u_{ij} > 0; \tag{I-71}$$

$$u_{i,j}D_x u_{i,j} = u_{i,j}\frac{u_{i+1,j}-u_{i-1,j}}{2h} + qu_{i,j}\frac{u_{i-1,j}-3u_{i,j}+3u_{i+1,j}-u_{i+2,j}}{3h}, \quad \text{if } u_{i,j}<0; \tag{I-72}$$

where parameter q controls the size of the modification. Similar schemes arise associated with the difference operators D_y and D_{yy} in Eqs. (I-65)-(I-67). The u,v boundary conditions are usually given in forms of Dirichlet (i.e., condition (I-74)) or Neumann conditions; the pressure boundary conditions are supplied by Eq. (I-68) given as an explicit form of Dirichlet condition from iteration to iteration. Specifically, we use (written in a semi-discretized form)

$$p^{\left(n+\frac{1}{2}\right)} = p^{\left(n-\frac{1}{2}\right)} + \Delta tc^2\left\{u_x+v_y + w_1(uu_x+vu_y+p_x-Re^{-1}(u_{xx}+u_{yy})) \right.$$
$$\left. w_2(uv_x+vv_y+p_y-Re^{-1}(v_{xx}+v_{yy}))\right\} \quad \text{on } \Gamma \tag{I-73}$$

as pressure boundary conditions which ensures the momentum and incompressibility equations satisfied. In Eq. (I-73), we use the one-sided derivatives with second-order approximations for the difference operators D_x, D_{xx}, D_y and D_{yy} whenever necessary. As shown in **Figure 1.1**, the dark region is where the velocities, pressure and their derivatives are approximated by finite differences with second-order accuracies for Eqs. (I-65)-(I-67), and the pressure boundary conditions in bright region are obtained by Eq. (I-73).

The overall solution procedure involves the following sequential steps:

i) construct the initial values of $u^{(0)}$, $v^{(0)}$ and $p^{(0)}$;

ii) solve Eq. (I-65) for advanced values of u at (n)th time step;

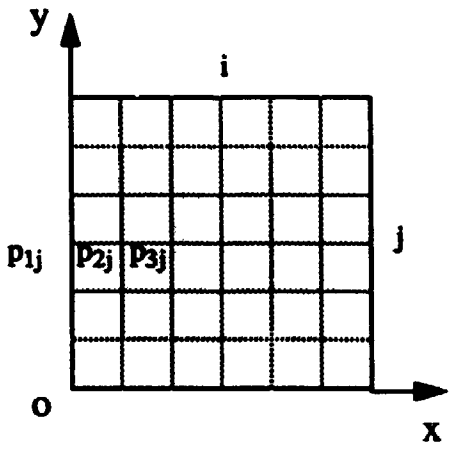


Figure 1.1 Uniform Discretized Solution Domain

iii) solve Eq. (I-66) for advanced values of v at (n)th time step;

iv) solve Eq. (I-67) for p in the field at (n+$\frac{1}{2}$)th time step;

v) obtain p at the boundary by Eq. (I-73) at the same step as (iv);

vi) repeat ii) - v) at (n+1)st time step until the steady state is reached.

The numerical scheme in (I-73) is stable for Δt small enough, and is entirely explicit. The coefficients w_1 , w_2 which are chosen as 1 or -1 in the present calculation. Their signs play roles similar to that of a parameter in upwind scheme, to diminish the accumulation of truncation errors so that the system remains stable from one iteration to another. The numerical experiments show that the signs of w_1 , w_2 are important at the corners where the residual of incompressibility is zero or remains as a constant, i.e., at the corner point (1,1) adjacent to both walls, $D_{1,1} = (D_x u_{1,1} + D_y v_{1,1}) = 0$;if $p_{1,1}^{(n+\frac{1}{2})} > p_{1,1}^{(n-\frac{1}{2})}$, then the signs of w_1 , w_2 have to be chosen to ensure $w_1 R u_{1,1} + w_2 R v_{1,1} > 0$, to prevent the computed values away from the real solutions.

Generally, the different signs will result in a different determined system that approaches to the Navier-Stokes system. Our understanding is that w_1,w_2 are arbitrary other than corner points as long as the numerical scheme is stable, since the residual $D_h(x_i, y_j)$ is considered as the dominant term (compared with residuals of momentum) in the equation which changes about two digits ahead of those of momentum residuals from iteration to iteration.

CHAPTER 3: THE COMPUTATIONAL BOUNDARY CONDITION METHOD

3.1 One Dimensional Illustration

The basic idea of this approach is the use of implicit boundary conditions in Navier-Stokes type equations. We will illustrate the method in two model problems which are related to stream function-vorticity and primitive variable forms of the Navier-Stokes equations.

The first model is a system of two second-order equations for two dependent variables ψ and ζ, which has been used by Dennis and Quartapelle (1989). We write

$$\zeta'' + g(x, \psi) \cdot \zeta' = f(x, \psi), \tag{I-74}$$

$$\psi'' = -\zeta, \qquad 0 \le x \le 1, \tag{I-75}$$

with boundary conditions for one variable ψ only

$$\psi(0) = \psi(1) = 0, \tag{I-76}$$

$$\psi'(0) = \psi'(1) = 0, \tag{I-77}$$

where prime denotes the differentiation with respect to x, the function f and g could either be given or depend upon one variable ψ in which case the problem carries nonlinear nature. No explicit boundary conditions are given for the variable ζ while the

boundary conditions for ψ are overspecified.

The differential system (I-74)-(I-77) is a determined system in the sense that four boundary constraints are given for two second order differential equations. However, when a standard finite difference approach is used to discretize this system as

$$D_{xx}\zeta_i + g_i \cdot D_x\zeta_i = f_i, \tag{I-78}$$

$$D_{xx}\psi_i = -\zeta_i, \qquad x_2 \le x_i \le x_{M-1}, \tag{I-79}$$

$$\psi_1 = \psi_M = 0, \tag{I-80}$$

$$D_x\psi_1 = D_x\psi_M = 0. \tag{I-81}$$

where D_x and D_{xx} are the first and second difference operators. The difference system (I-78)-(I-81) is not determined because no boundary conditions are available for ζ values on the physical boundaries, i.e. x_1 (=0) and x_M (=1). Here we refer the physical solution domain and boundaries as those for the original differential equations. If the discretization of differential equation for ζ at physical boundaries is considered as the boundary condition for ζ, it overdetermines the system.

Now we derive a determined system by our second new approach, the Computational Boundary Condition method. First of all, a mesh system with uniform grid size h and nodes $x_i = (i-1)\,h$, $i = 1, \ldots, M$ is set up on the physical domain, $0 \le x \le 1$. Then the domain from x_2 to x_{M-1} is defined as computational domain, i.e., one grid size into the physical domain. The computational boundaries are also moved one grid size in as x_2 and x_{M-1}. It is shown in **Figure 1.2.**

We should mention here that the computational domain is related to a certain

discretized system obtained from system (I-74)-(I-77) on computational domain is

$$D_{xx}\zeta_i + g_i \cdot D_x\zeta_i = f_i, \tag{I-82}$$

$$D_{xx}\psi_i = -\zeta_i, \qquad x_3 \le x_i \le x_{M-2}, \tag{I-83}$$

$$\psi_1 = \psi_M = 0, \tag{I-84}$$

$$D_x\psi_1 = D_x\psi_M = 0, \tag{I-85}$$

$$\zeta_2 = -D_{xx}\psi_2, \tag{I-86}$$

$$\zeta_{M-1} = -D_{xx}\psi_{M-1}, \tag{I-87}$$

which is now a determined system. Eqs.(I-86)-(I-87) derived from differential equation (I-75) for ψ provides the implicit boundary conditions for ζ while the discretization of boundary condition (I-84)-(I-85) furnishes the equations for ψ at the computational boundaries.

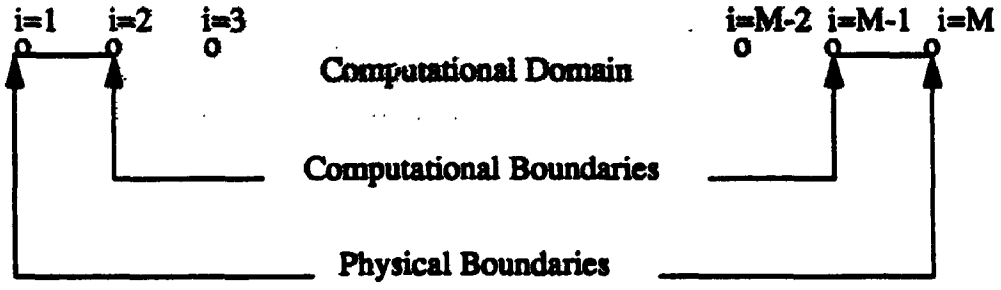


Figure 1.2 Illustration of Setting up the 1-D Computational Domain.

Our second model problem is given as

$$\psi_t + \psi'' + g(x, \psi) \cdot \zeta' = f(x, \psi), \tag{I-88}$$

$$\psi' = \varepsilon_1 \cdot \zeta_t + \varepsilon_2 \cdot \zeta, \tag{I-89}$$

with boundary conditions for one variable ψ only

$$\psi(0) = \psi(1) = 0 \quad \text{or} \quad \psi'(0) = \psi'(1) = 0 \tag{I-90}$$

where f and g are functions of x or one dependent variable ψ and, ε_1 and ε_2 are small constants. This one dimensional model is utilized as an analogue to the two or three dimensional Navier-Stokes equations with primitive variable formulation. The choice of small ε_1 here is for the purpose of simulating the artificial compressibility formulation when $\varepsilon_2 = 0$. On the other hand, small parameter ε_2 is to simulate the penalty method when $\varepsilon_1 = 0$. We are looking for the steady state solution of system (I-88)-(I-90), i.e., $\zeta_t = 0$, that essentially approaches to the solution of the differential system

$$\psi'' + g(x, \psi) \cdot \zeta' = f(x, \psi), \tag{I-91}$$

$$\psi' = \varepsilon_2 \cdot \zeta, \tag{I-92}$$

$$\psi(0) = \psi(1) = 0 \quad \text{or} \quad \psi'(0) = \psi'(1) = 0. \tag{I-93}$$

Actually system (I-91)-(I-93) could be reduced to one equation for ψ by eliminating the dependent variable ζ. However, it is useful to employ such a one dimensional model to analogue the two or three dimensional cases in which this elimination may not be possible in the uses of primitive variable formulation.

The finite difference approximation to system (I-88)-(I-90) can be written as

$$\frac{1}{\Delta t}(\psi_i - \psi_i^{n-1}) + D_{xx}\psi_i + g_i \cdot D_x\zeta_i = f_i, \tag{I-94}$$

$$D_x\psi_i = \frac{\varepsilon_1}{\Delta t}(\zeta_i - \zeta_i^{n-1}) + \varepsilon_2\zeta_i, \qquad x_3 \le x_i \le x_{M-2}, \tag{I-95}$$

$$\psi_1 = \psi_M = 0 \quad \text{or} \quad D_x\psi_1 = D_x\psi_M = 0, \tag{I-96}$$

$$D_{xx}\psi_2 + g_2 \cdot D_x\zeta_2 = f_2, \tag{I-97}$$

$$D_{xx}\psi_{M-1} + g_{M-1} \cdot D_x\zeta_{M-1} = f_{M-1}, \tag{I-98}$$

$$\varepsilon_2\zeta_2 = -D_x\psi_2, \tag{I-99}$$

$$\varepsilon_2\zeta_{M-1} = -D_x\psi_{M-1}. \tag{I-100}$$

As we have illustrated in the first model, the discretization forms (I-97)-(I-100) of the differential equation (I-91)-(I-92) at the computational boundaries x_2 and x_{M-1} provide boundary conditions for both dependent variables ψ and ζ. The difference from the first model problem is the uses of both equations at the computational boundaries. One-side difference formulae are introduced for the derivatives of ζ in (I-94) at computational boundaries. We should point out that the one-side derivatives of ζ in (I-94) at physical boundaries x_1 and x_M are not acceptable because the values of ζ at

those nodes are neither constrained by the difference equations nor by the boundary conditions. However, the derivatives at computational boundaries could be approximated by the one-side derivative in which the values of ζ could be determined by the difference Eqs. (I-99)-(I-100).

3.2 Application to Two Dimensional Model Problem in Stream Function-Vorticity Formulation

The computational boundary condition method developed in this thesis could be generalized in two or three dimensional cases. Their formulations are straightforward analog to one dimensional cases. In the following we will provide the details for two dimensional driven cavity problem in stream-function vorticity formulation.

The Navier-Stokes equations for the steady, incompressible viscous flow could be written in terms of vorticity ζ and stream-function ψ in the form of

$$\nabla^2 \psi = -\zeta, \tag{I-101}$$

$$\frac{1}{Re}\nabla^2 \zeta = \psi_y \zeta_x - \psi_x \zeta_y. \tag{I-102}$$

The boundary conditions generally associated with (I-101)-(I-102) are

$$\psi_s = d(s), \qquad \frac{\partial \psi}{\partial n} = n(s), \tag{I-103}$$

where n is the outward unit normal to the boundary and s is measured along it, ∇^2 is the Laplacian operator.

For the driven cavity problem, the physical solution domain is $\Omega\{(x,y), 0 \le x, y \le 1\}$. Let h be the uniform grid size, a mesh system with nodes (x_i, y_j), $x_i = (i\text{-}1)h, y_j = (j\text{-}1)h, i = 1, \ldots, M, j = 1, \ldots, N$. The computational domain is defined as $\Omega_h\{(x_i, y_j),\ x_3 \le x_i \le x_{M-2},\ y_3 \le y_j \le y_{N-2}\}$ which is illustrated in **Figure 1.3**. The system (I-101)-(I-103) is discretized on the computational solution domain Ω_h instead of physical domain Ω.

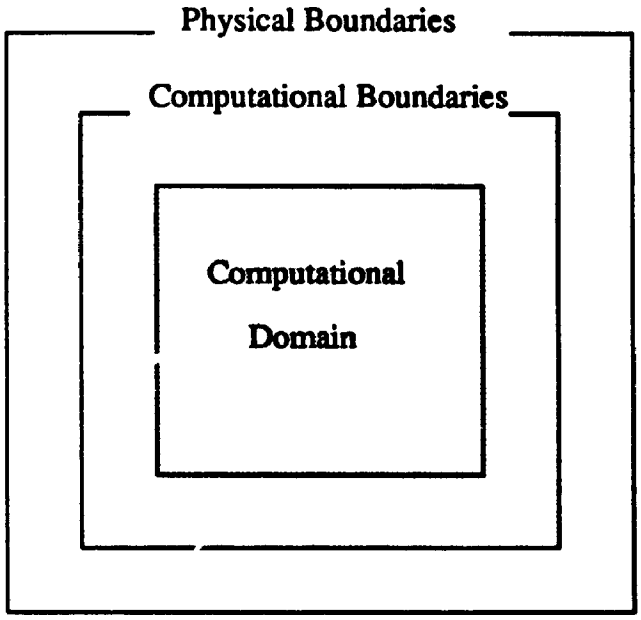


Figure 1.3. Illustration of Setting up the 2-D Computational Domain

The discretization system by using second-order central difference approach is two discretized equations for ζ and ψ inside the computational domain,

$$\frac{1}{Reh^2}(\zeta_{i-1,j} + \zeta_{i+1,j} + \zeta_{i,j-1} + \zeta_{i,j+1} - 4\zeta_{i,j}) = \qquad \text{(I-104)}$$

$$\frac{1}{4h^2}\left((\psi_{i,j+1}-\psi_{i,j-1})(\zeta_{i+1,j}-\zeta_{i-1,j})-(\psi_{i+1,j}-\psi_{i-1,j})(\zeta_{i,j+1}-\zeta_{i,j-1})\right),$$

$$\frac{1}{h^2}(\psi_{i-1,j}+\psi_{i+1,j}+\psi_{i,j-1}+\psi_{i,j+1}-4\psi_{i,j}) = -\zeta_{i,j}, \qquad \text{(I-105)}$$

$$i = 3, \ldots, M-2, \qquad j = 3, \ldots, N-2,$$

and the boundary conditions on the boundaries of computational domain

$$\zeta_{2,j} = -\frac{1}{h^2}\cdot(\psi_{3,j}+\psi_{2,j-1}+\psi_{2,j+1}-4\psi_{2,j}), \Leftrightarrow \zeta_{2,j} = -\nabla_h^2\psi_{2,j} \qquad \text{(I-106)}$$

$$\zeta_{M-1,j} = -\frac{1}{h^2}\cdot(\psi_{M-2,j}+\psi_{M-1,j-1}+\psi_{M-1,j+1}-4\psi_{M-1,j}),$$

$$\Leftrightarrow \zeta_{M-1,j} = -\nabla_h^2\psi_{M-1,j} \qquad \text{(I-107)}$$

$$\psi_{2,j} = \frac{1}{4}\psi_{3,j}, \qquad \Leftrightarrow D_x\psi_{1,j} = 0 \qquad \text{(I-108)}$$

$$\psi_{M-1,j} = \frac{1}{4}\psi_{M-2,j}, \qquad \Leftrightarrow D_x\psi_{M,j} = 0 \qquad \text{(I-109)}$$

$$j = 2, \ldots, N-1;$$

$$\zeta_{i,2} = -\frac{1}{h^2}\cdot(\psi_{i,3}+\psi_{i-1,2}+\psi_{i+1,2}-4\psi_{i,2}), \Leftrightarrow \zeta_{i,2} = -\nabla_h^2\psi_{i,2} \qquad \text{(I-110)}$$

$$\zeta_{i,N-1} = -\frac{1}{h^2}\cdot(\psi_{i,N-2}+\psi_{i-1,N-1}+\psi_{i+1,N-1}-4\psi_{i,N-1}),$$

$$\Leftrightarrow \zeta_{i,N-1} = -\nabla_h^2\psi_{i,N-1} \qquad \text{(I-111)}$$

$$\psi_{i,2} = \frac{1}{4}\psi_{i,3}, \qquad \Leftrightarrow D_y\psi_{i,1} = 0 \qquad \text{(I-112)}$$

$$\psi_{i,N-1} = \frac{1}{4}\psi_{i,N-2}-\frac{h}{2}, \qquad \Leftrightarrow D_y\psi_{i,N} = 1 \qquad \text{(I-113)}$$

$$i = 2, \ldots, M-1.$$

In the above, ∇_h^2 is a discrete laplacian operator. The boundary equations for ζ are

derived from (I-101) while ψ's boundary equations are the combinations of two equations in (I-103) by using three point difference for the Neumann conditions.

In the above, all the equations and boundary conditions have second-order accuracy, i.e., $O(h^2)$. For the computation of high Reynolds number flow, second-order upwind scheme for convective terms is employed. The convective terms involved in Eq.(I-104) are approximated by the type of difference scheme proposed by Ma (1983), i.e., the single-step upwind scheme for $u_{i,j}D_x\zeta_{i,j}$ is given:

$$u_{i,j}D_x\zeta_{i,j} = -(1-\alpha_{i,j})\frac{u_{i,j}}{2h}(\zeta^{n-1}_{i+1,j}-\zeta^{n-1}_{i,j})$$

$$-(1+\alpha_{i,j})\frac{u_{i,j}}{2h}(\zeta_{i,j}-\zeta_{i-1,j}), \qquad \text{if } u_{i,j}\le 0; \quad \text{(I-114)}$$

$$u_{i,j}D_x\zeta_{i,j} = -(1-\alpha_{i,j})\frac{u_{i,j}}{2h}(\zeta_{i+1,j}-\zeta_{i,j})$$

$$-(1+\alpha_{i,j})\frac{u_{i,j}}{2h}(\zeta^{n-1}_{i,j}-\zeta^{n-1}_{i-1,j}), \qquad \text{if } u_{i,j}\ge 0; \quad \text{(I-115)}$$

$$\text{where} \qquad u_{i,j} = D_y\psi_{i,j}.$$

A similar difference form associated with the convective term $v_{i,j}D_y\zeta_{i,j}$ could be given, where $v_{i,j} = D_x\psi_{i,j}$. The truncation error of such scheme is

$$\text{T.E.} = O(h^2, \alpha_{i,j}h), \qquad \text{(I-116)}$$

Here the parameter $\alpha_{i,j}$ is introduced to eliminate the highly oscillating behavior due to high Reynolds number. The values of $\alpha_{i,j}$ are chosen as the same order of the mesh size such that the truncation errors remain as the second-order, i.e, $\alpha_{i,j} = h$ throughout

the solution domain.

The system (I-104)-(I-113) using Computational Boundary Condition method can be solved with various difference schemes. In the present case we solve Eqs.(I-104)-(I-105) in domain Ω_h by standard ADI solution procedure (Peaceman and Rachford 1955): only terms associated with a particular coordinate direction are treated implicitly. To simplify the program, we retain the convective terms in Eq.(I-104) on the right hand side where the second-order central difference formula is kept. A coupled system of Poisson equations for ψ and ζ is solved by iterative solution procedure. For the Reynolds number higher than 1000, Ma's single-step upwind schemes (I-114) and (I-115) are employed in the vorticity transport equation to avoid the divergence. In this case, part of convective terms will be treated implicitly. This treatment is to stabilize the difference scheme while the diffusive terms become less important in the equation due to the high Reynolds flow. In any of these difference forms, the system (I-104)-(I-105) leads to the coupled system of tridiagonal matrices which could be solved by economical Thomas algorithm.

The general structure of the tridiagonal matrix is written as follows: (where L = M or N)

$$\begin{bmatrix} b_2 & c_2 & & & & \\ a_3 & b_3 & c_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{L-2} & b_{L-2} & c_{L-2} \\ & & & & & a_{L-1} & b_{L-1} \end{bmatrix} . \qquad (117)$$

It is always stable with no growth of rounding-off errors, if

i) $a_i < 0$, $c_i < 0$ and $b_i > 0$,

ii) $b_i > |a_{i+1} + c_{i-1}|$ for $i = 2, 3, ..., L-1$,

iii) $b_i > |a_i + c_i|$ for $i = 2, 3, ..., L-1$.

The condition i) and ii) ensure that the forward elimination is stable, while the condition i) and iii) ensure the back substitution is stable.

For solving the vorticity transport equation (I-104), over-relaxation is also adopted in the single-step upwind scheme to increase the convergence. The overall solution procedure involves the following sequential steps:

1. construct the initial values of ζ and ψ;
2. solve Eqs. (I-104)-(I-105) for x-sweep to obtain the advanced values of ζ and ψ;
3. solve Eqs. (I-104)-(I-105) for y-sweep to obtain the advanced values of ζ and ψ;
4. repeat step 2 and step 3 until the convergence criteria is reached.

3.3 Application to Two Dimensional Model Problem in Primitive Variable Formulation

In this section, we will generalize the computational boundary condition method to two dimensional cases for primitive variable formulation. It is a straightforward analog to one dimensional model. The details for the two dimensional driven cavity problem in primitive variable formulation will be provided as follows.

We write the steady state Navier-Stokes Equations and divergence equation in forms of the artificial compressibility method (Chorin 1967) in domain Ω and its

boundary Γ where ε is an arbitrary constant ($\varepsilon << 1$)

$$\mathbf{V}_t + \mathbf{V}\nabla\mathbf{V} + \nabla p - Re^{-1}\nabla^2\mathbf{V} = 0, \qquad \text{in } \Omega \tag{I-118}$$

and

$$\varepsilon p_t + \nabla \bullet \mathbf{V} = 0\,. \qquad \text{in } \Omega \tag{I-119}$$

The method could be applied for both Dirichlet and Neumann boundary conditions. Namely

$$\mathbf{V} = \mathbf{V}_\Gamma \quad \text{or} \quad \frac{\partial \mathbf{V}}{\partial n} = \mathbf{S}_\Gamma \qquad \text{on } \Gamma \tag{I-120}$$

Since we are considering results for two dimensional driven cavity problem, we will apply the Dirichlet boundary conditions in deriving the formulation. In cartesian coordinates, the system is written as

$$u_t - \frac{1}{Re}(u_{xx} + u_{yy}) + uu_x + vu_y + p_x = 0, \tag{I-121}$$

$$v_t - \frac{1}{Re}(v_{xx} + v_{yy}) + uv_x + vv_y + p_y = 0, \tag{I-122}$$

$$u_x + v_y + \varepsilon p_t = 0, \qquad 0 \le x, y \le 1 \tag{I-123}$$

$$u = v = 0, \qquad \text{at} \quad x = 0, 1 \quad \text{and} \quad y = 0,$$

$$u = 1,\ v = 0, \qquad \text{at} \quad y = 1. \tag{I-124}$$

Similar as in stream-function vorticity formulation, the two momentum equations and

the divergence equation are discretized on the computational solution domain,

$$\frac{1}{\Delta t}(u_{i,j}-u_{i,j}^{n-1})-\frac{1}{Reh^2}(u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}-4u_{i,j})$$

$$+\frac{u_{i,j}}{2h}(u_{i+1,j}-u_{i-1,j})+\frac{v_{i,j}}{2h}(u_{i,j+1}-u_{i,j-1})+D_x p_{i,j}^{n-1}=0, \qquad \text{(I-125)}$$

$$\frac{1}{\Delta t}(v_{i,j}-v_{i,j}^{n-1})-\frac{1}{Reh^2}(v_{i-1,j}+v_{i+1,j}+v_{i,j-1}+v_{i,j+1}-4v_{i,j})$$

$$+\frac{u_{i,j}}{2h}(v_{i+1,j}-v_{i-1,j})+\frac{v_{i,j}}{2h}(v_{i,j+1}-v_{i,j-1})+D_y p_{i,j}^{n-1}=0, \qquad \text{(I-126)}$$

$$\frac{\varepsilon}{\Delta t}(p_{i,j}-p_{i,j}^{n-1})+\frac{1}{2h}(u_{i+1,j}-u_{i-1,j})+\frac{1}{2h}(v_{i,j+1}-v_{i,j-1})=0, \qquad \text{(I-127)}$$

$$2\le i\le M-1, \quad 2\le j\le N-1$$

$$u_{1,j}=v_{1,j}=0, \qquad j=2,\ldots,N-1 \qquad \text{(I-128)}$$

$$u_{M,j}=v_{M,j}=0, \qquad j=2,\ldots,N-1 \qquad \text{(I-129)}$$

$$u_{i,1}=v_{i,1}=0, \qquad i=2,\ldots,M-1 \qquad \text{(I-130)}$$

$$u_{i,N}=1, \quad v_{i,N}=0, \qquad i=2,\ldots,M-1 \qquad \text{(I-131)}$$

where all the superscripts n for advanced time level are omitted. $D_x p_{i,j}$ and $D_y p_{i,j}$ are discretized by central difference except at the computational boundaries where one-side difference are used, i.e.,

$$D_x p_{2,j}=-\frac{3p_{2,j}-4p_{3,j}+p_{4,j}}{2h}, \qquad \text{(I-132)}$$

$$D_x p_{M-1,j} = \frac{3p_{M-1,j} - 4p_{M-2,j} + p_{M-3,j}}{2h}, \tag{I-133}$$

$$j = 2, \ldots, N-1;$$

$$D_y p_{i,2} = -\frac{3p_{i,2} - 4p_{i,3} + p_{i,4}}{2h}, \tag{I-134}$$

$$D_y p_{i,N-1} = \frac{3p_{i,N-1} - 4p_{i,N-2} + p_{i,N-3}}{2h}, \tag{I-135}$$

$$i = 2, \ldots, M-1.$$

Thus no boundary condition is needed for pressure at the physical boundary. The boundary conditions for velocity components are given at physical boundaries. Again central difference is used for convective terms. It is replaced by second-order upwind difference at high Reynolds number calculations.

The system (I-125)-(I-131) using Computational Boundary Condition method can be solved with various difference schemes. In this case, the standard ADI solution procedure (Peaceman and Rachford 1955) also could be used to solve the momentum equations. The higher-order upwind scheme (Fletcher 1988) (based on four-point difference scheme) is introduced for the convective terms, i.e.,

$$u_{i,j} D_x u_{i,j} = u_{i,j} \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$

$$+ q u_{i,j} \frac{u_{i-1,j} - 3u_{i,j} + 3u_{i+1,j} - u_{i+2,j}}{3h}, \qquad \text{if} \qquad u_{i,j} \le 0; \tag{I-136}$$

$$u_{i,j}D_x u_{i,j} = u_{i,j}\frac{u_{i+1,j}-u_{i-1,j}}{2h}$$

$$+ qu_{i,j}\frac{u_{i-2,j}-3u_{i-1,j}+3u_{i,j}-u_{i+1,j}}{3h}, \qquad \text{if} \qquad u_{i,j}\geq 0; \qquad \text{(I-137)}$$

where parameter q controls the size of modification. Similar schemes arise associated with other convective terms. The overall solution procedure involves the following sequential steps:

1. construct the initial values of u,v and p;
2. solve Eqs.(I-125)-(I-126) for x-sweep to obtain the advanced values of u and v;
3. solve Eqs.(I-125)-(I-126) for y-sweep to obtain the advanced values of u and v;
4. solve Eq.(I-127) for advanced value of p;
5. repeat step 2-step 4 until the steady state is reached.

CHAPTER 4: NUMERICAL RESULTS AND COMPARISONS

All numerical experiments were performed on uniform grids by means of Zero Perturbation Method and Computational Boundary Condition method described in chapter three and four respectively. In the first instance we provide a one dimensional example corresponding to the first model in stream-function vorticity formulation. The results are given based on uses of CBC method. The two dimensional examples are given to solve for the driven cavity problems using both Zero Perturbation Method and Computational Boundary Condition methods. For two dimensional illustrations, calculations of convergence rate are made on the free surface problem with primitive variable formulation in which the pressure singularity has been smoothed out by using a regularized driven velocity profile. The methods are also used to solve for the classical driven cavity problem. Comparisons of numerical accuracy are made with the available results obtained by previous researchers.

4.1 Numerical Results on One Dimensional Model Problem

First we start with a simple example which models the biharmonic equation in one space dimension which is solved by CBC method. For this we consider Eqs. (I-74)-(I-77) with the exact solutions

$$\psi(x) = \frac{1}{2}x^2 - \frac{3}{4}x^3 + \frac{1}{4}x^5, \tag{I-138}$$

$$\zeta(x) = 1 - \frac{9}{2}x + 5x^3, \tag{I-139}$$

which satisfy the following equations with boundary conditions (I-76)-(I-77)

$$\psi'' = \zeta(x), \tag{I-140}$$

$$\zeta'' = 30x, \tag{I-141}$$

by choosing $f(x)$=30x and $g(x)$=0 in the one dimensional stream-function vorticity illustration. We may obtain the analytical solutions by finding a solution for ζ of (I-141), and then determine ψ afterwards by solving (I-140) subject to the boundary conditions.

The primary goal in this test is to assess the accuracy and the convergence rates of the numerical approximation of ζ and ψ compared with the analytical solutions using CBC method. The numerical calculations were carried out using exactly the same second-order finite difference formulae (I-82)-(I-83) with the boundary conditions (I-84)-(I-87).

The computed results are shown in **Table I-I** with h=0.05 and h=0.1. It could be confirmed that the results are obtained within expected digit accuracy of second-order. To determine the convergence rates, we examine the root mean square errors (E_2) as function of grid sizes h by computing

$$E_2 = \frac{\|F_a - F_c\|_2}{\|F_a\|_2}, \tag{I-142}$$

where $E_2 = O(h^{\alpha})$. In the above, F_a represents the values of the exact solutions of ζ and ψ; F_c represents their computational solutions; $\|\,.\,\|_2$ is the discrete L_2-norm on the discretized grid points and

$$\|F\|_2 = \frac{1}{l^2}\sqrt{\sum_{i=1}^{l}|F_i|^2}, \qquad \text{(I-143)}$$

where l being the number of points where the values are calculated. As shown in **Table I-I**, the convergence rate α of ζ and ψ obtained by two different grid sizes on the overlapped points of computed results are predicted second order.

TABLE I-I

Convergence Rate Analysis of 1-D model Problem

	Globle Rate ($0 < x < 1$)	Point Rate ($x = 0.5$)	Computed Values of Diff. Grids ($x = 0.5$)		
			N=11	N=21	Exact
ψ	1.88	1.88	-0.0338	-0.0376	-0.0391
ζ	1.94	1.90	-0.5700	-0.6103	-0.6250

Note: The globle rate is calculated on L_2 norm;

In this case, it gave approximately the same results as the integral constraint method, i.e., within a second order accuracy. However, the present formulations are much simpler and could be easily extended to two and three dimensional cases.

4.2 Numerical Results of Zero Perturbation Method

Three numerical test problems were performed on fully nonstaggered grids by means

of zero perturbation method described in chapter two: a Stokes flow problem with analytical solution, the free surface problem with a driven velocity at the bottom and a classical driven cavity problem. Comparisons of accuracy, convergence rate are made for these test problems. The domain for all the test problems is $0 \le x \le 1, 0 \le y \le 1$.

Analytical Stokes Flow. First, the accuracy of the present method has been tested on the exact solution

$$u(x,y) = -\cos(a\pi x)\sin(a\pi y) \tag{I-144}$$

$$v(x,y) = \sin(a\pi x)\cos(a\pi y) \tag{I-145}$$

$$p(x,y) = -2a\pi \sin(a\pi x)\sin(a\pi y) \tag{I-145}$$

$$f = 4a^2\pi^2\cos(a\pi x)\sin(a\pi y), \tag{I-146}$$

which satisfy the following equations for the Stokes flow

$$u_{xx} + u_{yy} = p_x + f \tag{I-147}$$

$$v_{xx} + v_{yy} = p_y \tag{I-148}$$

$$u_x + v_y = 0. \tag{I-149}$$

The purpose of this test is to assess the accuracy of the numerical approximation of the velocities and pressure compared with analytical solutions. The calculations are carried out from uniform grid size h = 0.05 to h = 0.0125 with a $=\frac{1}{2}$.

Figure 1.4 displays the maximum errors (E_∞) and the relative errors ($\bar{E}$) versus the number of grid points N. It could be confirmed that the results were obtained within expected digit accuracies of second-order. For example, the computed velocity components u_c, v_c and pressure p_c along the horizontal centre line of the cavity

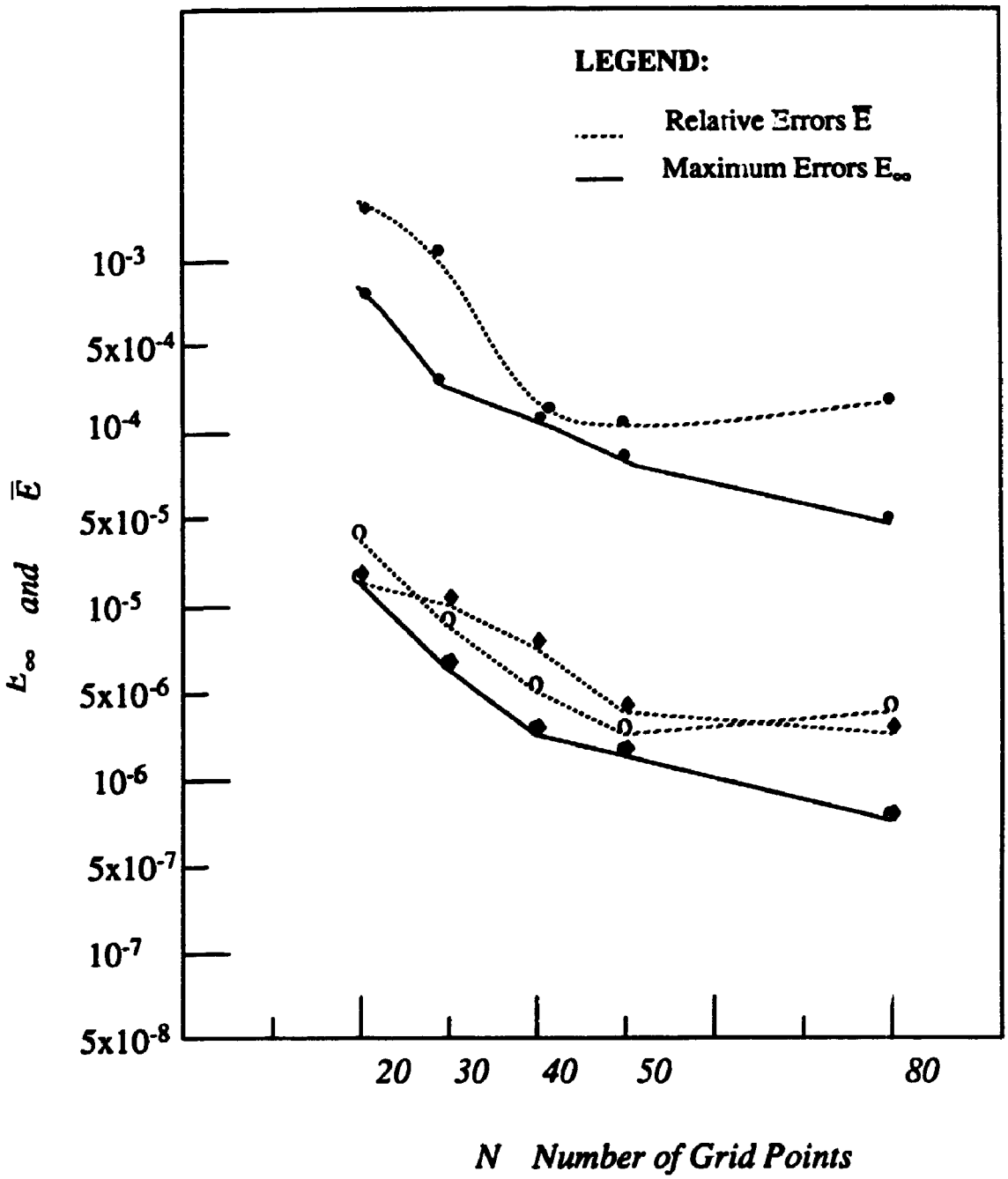


Figure 1.4 Truncation Errors Convergence As a Function of the Number of Grid Points N; Circles -- Errors of u Velocity; Black Diamond -- Errors of v Velocity; Black Circles -- Errors of Pressure.

located at (0.8,0.5) with uniform mesh h = 0.02 are

$$u_c = -0.2185080, \quad v_c = 0.6724985, \quad p_c = -2.112716 \qquad \text{(I-150)}$$

and the corresponding analytical solutions are

$$u_a = -0.2185082, \quad v_a = 0.6725006, \quad p_a = -2.112806, \qquad \text{(I-151)}$$

which have approximately 6-digit accuracy for the velocities and 4-digits for the pressure. All the errors of u,v and p between (I-150) and (I-151) are roughly within the truncation errors of $O(0.02^2)$. One may notice that the maximum error in 80×80 grid is higher than 50×50 grid in the present case. This may attribute to the higher frequency of residual function in 80×80 grid, which results in extrapolated approximation being less accurate than 50×50 grid, since the derivatives of residual function in 80×80 grid may have higher values.

It should be pointed out that the errors of pressure are about one and half order lower than the velocities in the present test problem as illustrated in **Figure 1.4**. This may be attributed to: 1) the variations of truncation errors of pressure are greater than the velocities; these could be confirmed by their analytical solutions; 2) the drawback of finite difference method, in which a parabolic approximation for the second derivative of velocities (based on three points) and a linear approximation (based on two points in the field) for the first derivative of pressure results in a better simulation for velocities than pressure in Stokes flow.

The determination of the convergence rates could be done by examining the

root mean square errors (E_2) as a function of grid sizes h by computing E_2 where $E_2 = O(h^{\alpha})$. As shown in **Table I-II**, the convergence rates α of velocities and pressure obtained by two different grid sizes on the overlapped points of computed results are the predicted 2.0.

TABLE I-II

E_2 Convergence Rates for Test Problem with an Exact Solution

	Conv. Rates for Velocities		Conv. Rates for Pressures		
Grid Pt. No.	u	v	P	P_f	P_B
G20-G40	1.99	1.98	1.93	1.90	1.97
G30-G40	1.97	1.98	2.04	2.06	2.02
G30-G50	2.00	1.99	1.94	1.96	1.92
G30-G80	1.99	1.99	1.94	1.95	1.92
G50-G80	1.98	1.99	1.94	1.88	1.99

Note: u, v represent the convergence rates of velocity components in the solution domain; P, P_f and P_B are the values in domain, field and boundaries respectively.

The convergence rates for the pressure at the boundaries are also presented in

this table. The velocity rates of convergence are approximately ±0.02 accurate where the pressure rates are about ±0.05 accurate. This could be again attributed to the different variations of their truncation errors. Therefore, the results provide an excellent agreement with the theory discussed in chapter two.

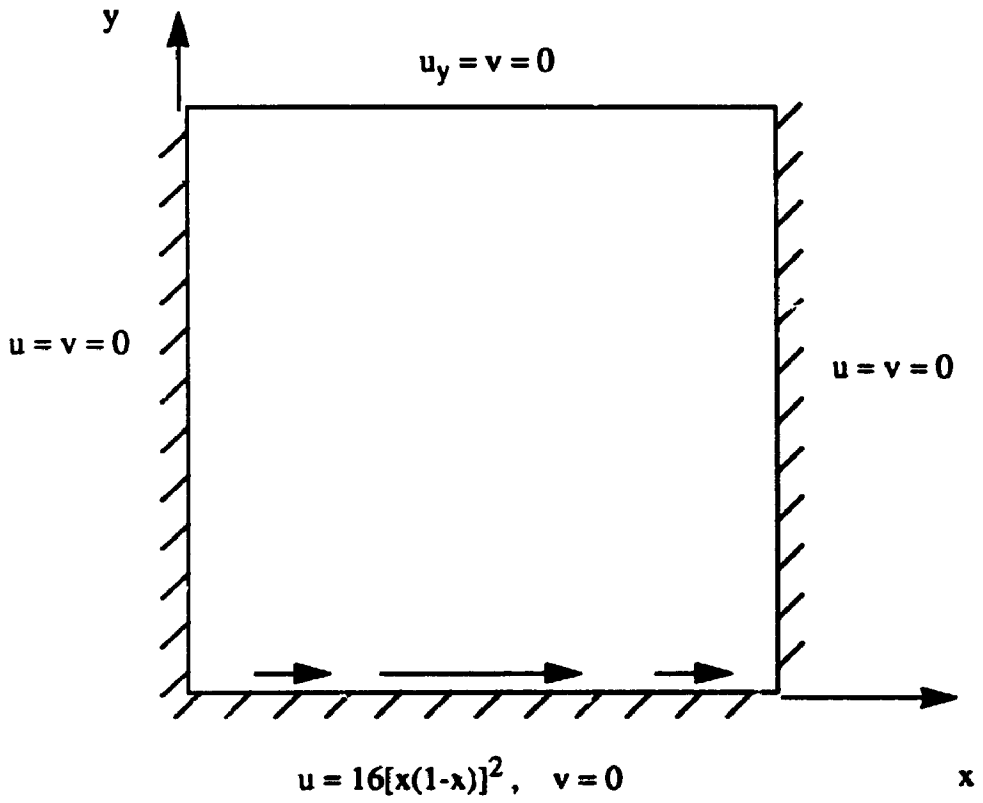


Figure 1.5 Geometry and Boundary Conditions For Free Surface Problem

Free Surface Flow. The second test problem for the zero perturbation method is a free surface problem. **Figure 1.5** shows the geometry and boundary conditions for the flow in a cavity with the appropriate nomenclature. There is a driven velocity $u=16[x(1-x)]^2$ at the bottom of the cavity and a free surface at the top. The purpose of this test is to assure the convergence rates of pressure and velocity approximations for

the viscous flow with Reynolds number. The analytical solutions are not available. For this case, the first order singularities are smoothed out by the type of boundary conditions specified for the driven velocity u at the bottom. So we could avoid the standard driven cavity problem where upper corner singularities may deteriorate the convergence rates. The convergence rates are evaluated for u,v and p from the numerical results obtained by three different grid sizes h_1, h_2 and h_3 by the following formula (written as F for u, v or p variables)

$$\frac{h_2^\alpha - h_1^\alpha}{h_2^\alpha - h_3^\alpha} = \frac{\left\| F_{h_2} - F_{h_1} \right\|_2}{\left\| F_{h_2} - F_{h_3} \right\|_2} \quad . \tag{I-152}$$

TABLE I-III

E_2 Convergence Rates for the Free Surface Test Problem

Grid Point Number	Re = 50			
	u	v	P	P_B
G20 - G30 - G40	2.17	1.90	1.91	1.92
G20 - G40 - G80	2.04	2.00	1.92	1.85
G30 - G50 - G80	1.93	2.04	1.98	2.01
G30 - G50 - G100	1.91	2.05	2.07	1.98

Note: The first column is three different grid sizes in computing the convergence rates; P_B stands for the convergence rates of pressure at the boundaries.

TABLE I-IV

Pointwise, Free Surface Conv. Rates and Its Computed Values
(Re =50)

Location (x,y) of the Point	Pressure Conv. Rates P	Computed Values at Diff. Grids		
		N = 30	N = 50	N = 100
(0, 0)	1.80	*P=-.13307*	*P=-.13578*	*P=-.13707*
(1, 0)	1.79	*P=0.28064*	*P= 0.28428*	*P= 0.28600*
(0, 1)	2.17	*P=.028631*	*P=.029575*	*P=.029936*
(1, 1)	2.14	*P=.016888*	*P=.017702*	*P=.018017*
(0.5,0.5)	2.18	*u=-0.14975* *v=-0.02832*	*u=-0.15156* *v=-0.02818*	*u=-0.15235* *v=-0.02815*
(0.5,1)	2.25	*u=-0.06613* *P=0.021482*	*u=-0.06601* *P=0.02253α*	*u=-0.06600* *P=0.022855*
Surface Line	2.26			

Note: The computed values of pressure are normalized by assigning p = 0 at (0.5,0.5) which is the centre of the cavity; The last line is a global conv. rate at free surface.

Table I-III summarizes the calculation of α for u, v and p at Re=50 for the free surface flow in the cavity. The results were obtained with the number of grid points N = 20, 30, 40, 50, 80 and 100. The computed convergence rates for u,v and p once again agree with the theoretical prediction. In regard to the convergence rates of computed pressure at the boundary, it can be seen in the table that both accuracies are the same for pressure in the field and at the boundary. Therefore, this experiment confirms the theory of equal rates of convergence for both field and boundary discussed in Chapter 2.

Additional convergence analyses are presented in **Table I-IV** and illustrate the point-wise convergence rates for the pressure at the four corner singular points and along the free surface line. For comparison, their computed values are also given. As seen in the table, the computed pressures are convergent with second-order accuracy at the corner singular points where the actual pressures are not infinite. So we may conclude that the present algorithm has guaranteed the consistent convergence rates with order of two. In the next test, we will also demonstrate that the method offers an effective approximation to the Navier-Stokes equations, even with infinite pressures at the singularities.

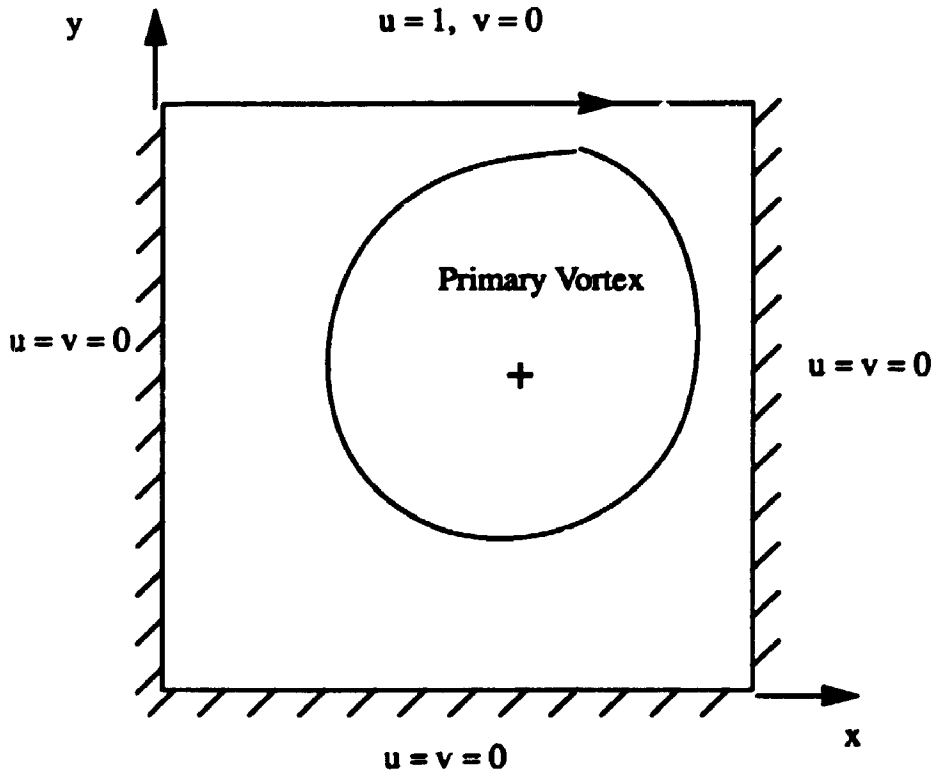


Figure 1.6. Geometry of Driven Cavity Problem

Driven Cavity Flow. The third test problem is the viscous flow in a driven

cavity. **Figure 1.6** shows the boundary conditions for the cavity flow with the appropriate nomenclature. At the top of the cavity, the driving velocity u is a constant (u = 1) along x-direction which generates several standing vortices inside cavity. The problem is generally recognized as a standard test for the efficiency and accuracy in the numerical simulation of Navier-Stokes equations, since it contains the general characteristic of elliptic and nonlinear nature of many flow problems. The accurate results for the flow have been published by many authors in the last ten years, i.e., the results of Ghia *et al* (1982) are in good agreement with the studies by Schreiber and Keller (1984) for Reynolds number up to 10,000; both results are considered as standard solutions for this driven cavity flow problem. However, most of the accurate solutions are obtained by stream function-vorticity formulation. The results of primitive variable formulation are not quite satisfactory, especially, for the accurate solutions of pressure field. The present technique is able to provide the accurate solutions for the velocities as well as the pressure. We understand that the driven cavity flow contains the infinite values of pressure at the two upper corners which would affect the accuracies of these results. Therefore, the presented results in the following are chosen away from the singularities which still maintain four digit accuracies for the velocities and three digits for the pressure field on a 129×129 grid.

The computations are carried out for Re=100, Re=1,000, Re=5,000 and Re=10,000 with a 129×129 grid. The results on 129×129 grid are computed for its convenience to compare with available results, i.e., Ghia *et al.* For Re=100 and Re=1,000, the compressibility parameter c = 1 and time step $\Delta t = 0.005$, $\Delta t = 0.002$ were used respectively. For Re=5,000, a smaller time step compared with Re=1,000 (i.e., $\Delta t = 0.001$) was required to stabilize the difference scheme from iteration to

iteration, In this case, the

TABLE I-V

Comparisons of S ream-Function and Vorticity at Center of Primary Vortex for Different Reynolds Numbers

Minimum Stream-Function ψ_c and Prime Vorticity ξ_c:

Re	*Present*		*Ghia et al*		*Schreiber & Keller*	
	ψ_c	ξ_c	ψ_c	ξ_c	ψ_c	ξ_c
100	**-0.1033**	**-3.163**	**-0.1034**	**-3.166**	**-0.1033**	**-3.182**
	(129×129)		(129×129)		(121×121)	
1000	**-0.1159**	**-2.025**	**-0.1179**	**-2.050**	**-0.1160**	**-2.026**
	(129×129)		(129×129)		(141×141)	
5000	**-0.1130**	**-1.842**	**-0.1190**	**-1.860**	- -	
	(129×129)		(257×257)		- -	
10000	**-0.1060**	**-1.793**	**-0.1197**	**-1.881**	**-0.1028**	**-1.622**
	(129×129)		(257×257)		(180×180)	

integration time for momentum equations was performed from zero to t = 60 which takes about 30,000 iterations to reach the steady state. The accuracies of the present method were assessed by available results in the literature. Table I-V clearly shows that the present results for prime vortex and minimum stream function have at least three

digits in agreement with the results obtained by previous investigators (approximately the same grid sizes).

TABLE I-VI

Comparisons of u-Velocity along Vertical Line through Cavity Center $(\frac{1}{2}, \frac{1}{2})$

Grid Point Number	Location y	Re = 100		Re = 1000		Re = 5000	
		Present	*Ghia et al*	*Present*	*Ghia et al*	*Present*	*Ghia et al*
129	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
126	0.9766	0.8431	0.8412	0.6511	0.6593	0.4354	0.4822
125	0.9688	0.7911	0.7887	0.5667	0.5749	0.4183	0.4612
124	0.9610	0.7401	0.7372	0.5034	0.5112	0.4171	0.4599
123	0.9531	0.6905	0.6872	0.4593	0.4660	0.4202	0.4604
110	0.8516	0.2358	0.2315	0.3282	0.3330	0.3133	0.3356
95	0.7344	0.0040	0.0033	0.1840	0.1872	0.1858	0.2009
80	0.6172	-0.1386	-0.1364	0.0547	0.0570	0.0707	0.0818
65	0.5000	-0.2086	-0.2058	-0.0618	-0.0608	-0.0369	-0.0304
59	0.4531	-0.2134	-0.2109	-0.1070	-0.1065	-0.0787	-0.0740
37	0.2813	-0.1572	-0.1566	-0.2772	-0.2781	-0.2256	-0.2286
23	0.1719	-0.1014	-0.1015	-0.3774	-0.3829	-0.3137	-0.3305
14	0.1016	-0.0642	-0.0643	-0.2843	-0.2973	-0.3865	-0.4044
10	0.0703	-0.0465	-0.0478	-0.2091	-0.2222	-0.3850	-0.4364
9	0.0625	-0.0419	-0.0419	-0.1893	-0.2020	-0.3675	-0.4290
8	0.0547	-0.0371	-0.0372	-0.1692	-0.1811	-0.3420	-0.4117
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table I-VI and **Table I-VII** show the u velocity distributions along the vertical line and v along the horizontal line through the geometric center of the cavity respectively in the cases of Re=100, Re=1,000 and Re=5,000 on 129×129 grid. Comparisons of u

along the vertical line and v velocities along the horizontal line have shown that the results do not quite agree for Reynolds number greater than 5000. The present results differ from the stream functions and vorticities obtained by Ghia *et al* (1982). As shown in **Table I-VI** and **Table I-VII**, it has about two to three digits in agreement

TABLE I-VII

Comparisons of v-Velocity along Horizontal Line through Cavity Center $(\frac{1}{2},\frac{1}{2})$

Grid Point Number	Location x	Re = 100		Re = 1000		Re = 5000	
		Present	*Ghia et al*	*Present*	*Ghia et al*	*Present*	*Ghia et al*
129	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
125	0.9686	-0.0622	-0.0591	-0.2147	-0.2139	-0.4199	-0.4977
124	0.9609	-0.0778	-0.0739	-0.2766	-0.2767	-0.4852	-0.5507
123	0.9531	-0.0932	-0.0886	-0.3359	-0.3371	-0.5075	-0.5541
122	0.9453	-0.1083	-0.1031	-0.3896	-0.3919	-0.4978	-0.5288
117	0.9063	-0.1768	-0.1691	-0.5118	-0.5155	-0.3892	-0.4144
111	0.8594	-0.2331	-0.2245	-0.4204	-0.4267	-0.3430	-0.3621
104	0.8047	-0.2528	-0.2453	-0.3137	-0.3197	-0.2863	-0.3002
65	0.5000	0.0575	0.0545	0.0256	0.0253	0.0087	0.0095
31	0.2344	0.1790	0.1753	0.3185	0.3224	0.2554	0.2728
30	0.2266	0.1788	0.1751	0.3267	0.3308	0.2626	0.2807
21	0.1563	0.1642	0.1608	0.3653	0.3710	0.3285	0.3537
13	0.0938	0.1259	0.1232	0.3205	0.3263	0.3895	0.4295
11	0.0781	0.1114	0.1089	0.2980	0.3035	0.3894	0.4365
10	0.0703	0.1032	0.1009	0.2847	0.2901	0.3833	0.4333
9	0.0625	0.0945	0.0923	0.2695	0.2749	0.3729	0.4245
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

with the results of Ghia *et al* (1982) in cases of Reynolds number less than 1000. This

may be attributed to the differences between the formulations of primitive variables in the present method and stream function-vorticities of Ghia *et al* (1982) for solving the Navier-Stokes equations, i.e., the corner singularities may affect the accuracies of the results differently in these two formulations. It should be pointed out that, with the present method in primitive variable formulations, we can obtain not only the correct results of velocities and pressure but also the satisfactory results for stream function and vorticity field (within expected accuracy). This indicates that the advantage of present method may be better accuracy.

4.3 Numerical Results of Computational Boundary Condition Method

We solved the viscous flow in a driven cavity in forms of both stream-function vorticities and primitive variables. The details could be found in the previous section.

Results on stream-function vorticities. The computations in this formulation are carried out for Re=100, Re=1000 and Re=10000 with a 129×129 grid to compare with the available results. For Re=1000 and Re=10000, the upwind scheme is used to obtain the stability. **Table I-VIII** clearly shows that the present results for primary vortex and minimum streamline have at least three digits in agreement with the results obtained by previous investigators.

The calculations are also made on 65×65, 81×81 and 101×101 mesh points for Re=100, and 1000 to calculate the convergence rate.**Table I-IX** presents the convergence rates on the subsequential solutions that approach to the actual solution. The stream-function and vorticity values have exactly second-order rate of convergence to the solution.

TABLE I-VIII

Comparisons of Stream-Function and Vorticity at Center of Primary Vortex for Different Reynolds Numbers

(Stream-function vorticity formulation)

Minimum Stream-Function ψ_c and Prime Vorticity ξ_c:

Re	*Present*		*Ghia et al*		*Schreiber & Keller*	
	ψ_c	ξ_c	ψ_c	ξ_c	ψ_c	ξ_c
100	**-0.1034**	**-3.1650**	**-0.1034**	**-3.166**	**-0.1033**	**-3.182**
	(129×129)		(129×129)		(121×121)	
1000	**-0.1167**	**-2.0290**	**-0.1179**	**-2.050**	**-0.1160**	**-2.026**
	(129×129)		(129×129)		(141×141)	
10000	**-0.1129**	**-1.7411**	**-0.1197**	**-1.881**	**-0.1028**	**-1.622**
	(129×129)		(257×257)		(180×180)	

Results on primitive variables. The computations of this formulation are carried out for *Re*=100 and *Re*=1000 with a 129×129 grid for its convenience to compare the available results. In the convective terms, the higher-order upwind scheme discussed in Section 3 is applied to the case of *Re*=1000, and the central difference formula is kept for *Re*=100. For *Re*=100, the compressibility parameter ε=1 and time step Δt=0.01 were used. For *Re*=1000, a smaller time step (Δt=0.005) compared with *Re*=100 was required to stabilize difference scheme from iteration to iteration. In the case, the integration time for momentum equations was performed from t=0 to t=70

TABLE I-IX

Convergence Rates of Solution for Stream Function-Vortivities

Grid Point Number	Re	Convergence Rates	
		ψ	ζ
G60-G80-G100	100	2.799	2.078
G60-G80-G100	1000	3.566	2.060

TABLE I-X

Comparisons of Stream-Function and Vorticity at Center of Primary Vortex for Different Reynolds Numbers

(Primitive Variable formulation)

Minimum Stream-Function ψ_c and Prime Vorticity ξ_c:

Re	*Primitive*		*Stream Function*		*Kim and Moin*	
	ψ_c	ξ_c	ψ_c	ξ_c	ψ_c	ξ_c
100	**-0.1033**	**-3.1632**	**-0.1034**	**-3.1650**	**-0.1034**	**-3.177**
	(129×129)		(129×129)		(65×65)	
1000	**-0.11640**	**-2.0328**	**-0.1167**	**-2.0290**	**-0.1160**	**-2.026**
	(129×129)		(129×129)		(65×65)	

which takes about 14000 iterations to reach the steady state. The accuracies of the solutions were assessed by the available results in the literature, i.e., Kim and Moin (1985), and by our results obtained from stream-function vorticity formulation.

Table I-X presents the computed values of primary vortex and minimum stream-function in comparison with our results obtained from stream-function vorticity formulation, and Kim and Moin's results made by projection method with staggered grid. This illustration clearly shows the agreement with our stream-function vorticity solutions as well as the results of Kim and Moin (1985). These results provide the assurance of our new method to be extended to the three dimensional cases, and also the superior accuracy of this new algorithm. **Table I-XI** provides the computed convergence rates for Re=100, and Re=1000 obtained by grid sizes 61×61,81×81 and 101×101. Once again, it agrees perfectly with our theoretical prediction which is second-order accuracy.

TABLE I-XI

Convergence Rates of Solution for Primitive Variables

Grid Point Number	Re	Convergence Rates		
		u	v	p
G60-G80-G100	100	1.922	2.034	1.963
G60-G80-G100	1000	2.035	2.035	1.879

Finally, **TABLE I-XII** provides additional information of the efficiency and

accuracy of present algorithm of the CBC method. The comparisons have been made with SIMPLE algorithm on SUN workstation for grid size equal to 61×61 and Reynolds number 100. Both calculations were done using primitive variable formulation. The evidence in **TABLE I-XII** indicates that the CBC method has far less CPU times, programming codes as well as better accuracy in comparison with SIMPLE method.

TABLE I-XII

The Comparisons of CBC and SIMPLE method

Re =100, Grid = 61×61	CBC method	SIMPLE method[1]
FORTRAN Codes	250	1200
Primary Vortex	-3.134914	-3.015492
Minimum Streamline	-0.102511	-0.0850
CPU time (sec.)	312.5	1907.5

Note: Both convergence criteria for the present calculations was $|\nabla \bullet V| < 5\times10^{-5}$. SIMPLE method uses staggered grids. Both calculations were made on SUN microsystem Sparc 1^{+}.

[1]SIMPLE method was programmed by Mr. Huaxiong Huang, an engineering student at the University of British Columbia.

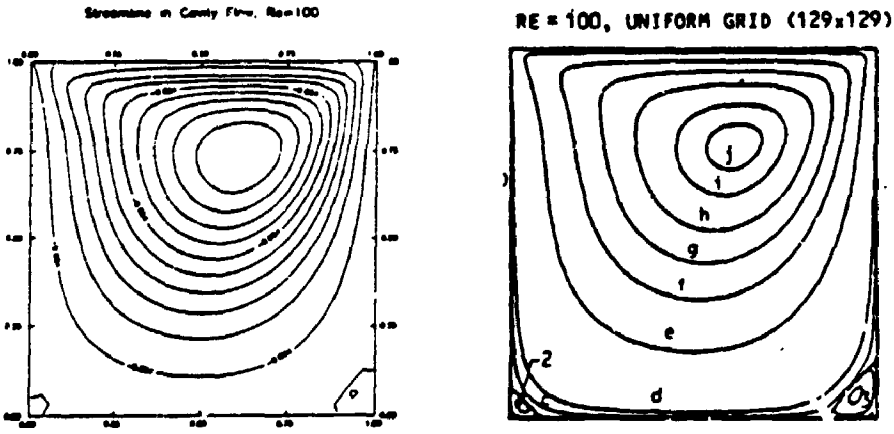
4.4 The Characteristics of the Driven Cavity Flow

The driven cavity problem is generally considered to be a standard test for the numerical algorithms since it contains a simple geometry and the boundary conditions also can be easily implemented for the algorithm. This problem characterizes the general natural of many flow problems. For example, this problem reflects the elliptic and nonlinear properties in the Navier-Stokes equations. Although there are a number of numerical studies made for this problem, the difficulties still remain in the accurate simulation of high Reynolds flows. In this case, the flow is highly unstable.

Numerous results have been obtained with finite difference, finite element and spectral approaches to this problem for the last two decades. A number of novel techniques have been employed to obtain the accurate solutions for Reynolds number up to 10,000. These excellent algorithms include the techniques of Benjamin and Denny (1979), Ghia *et al* (1982), Schreiber and Keller (1984) and Kim and Moin (1985). The first three methods use stream function-vorticity formulation while the last one using primitive variable formulation. The accurate results were obtained by Ghia *et al* (1982) who employed the second-order form of implicit vorticity condition in their efficient multigrid solution procedure. Their results for the driven cavity flow have been considered as standard solutions for a decade. The idea of Schreiber and Keller (1984) is to solve the biharmonic form of stream-function equation by eliminating the vorticity from the vorticity transport equation. The problem of vorticity boundary condition has been completely eliminated by such an implementation. Their results seem in good agreement with the studies of Ghia *et al* (1982). The coordinate transformation was selected by Benjamin and Denny (1979) in such way that a greater density of nodal points in the highly viscous boundary layer and secondary vortex (corner) regions than

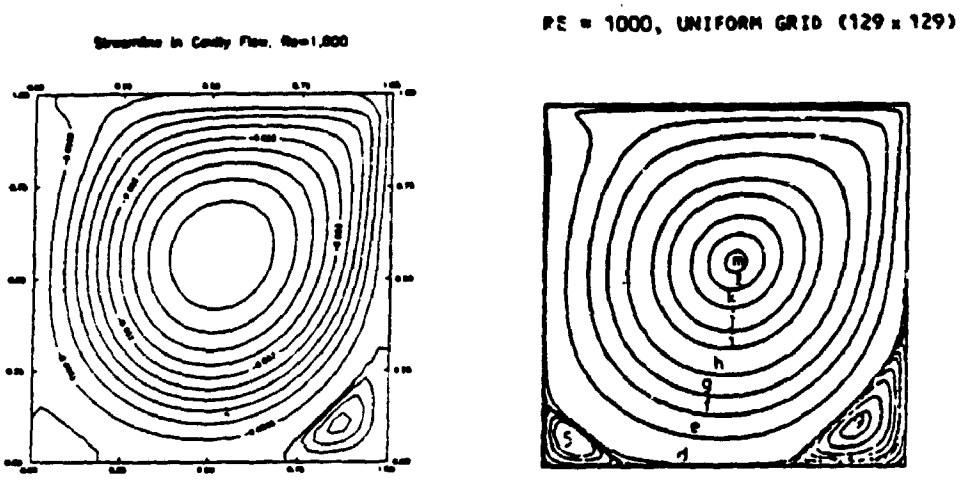
in the inviscid core. Their results, therefore, appear quite favorable for application to high Reynolds number flows. The results of Kim and Moin (1985) were less accurate than those using the stream function-vorticity formulation of the previous investigators. It uses a fractional step method combined with the idea of a staggered grid. Their results were still considered to be reasonable for solving the problem in primitive variable formulation. At least, it proves the concept of fractional step method to be correct.

The results for computed stream function and vorticity contours in the driven cavity flow are presented in **Figure 1.7** through **Figure 1.14** for Reynolds numbers 100, 1000,5000, 10000. The computed results are compared with the available results of the previous investigators, i.e., Ghia *et al* (1982), Schreiber and Keller (1984), in order to validate of the numerical procedure. The stream function and vorticity contours are



Present Results **Results of Ghia *et al* (1982)**

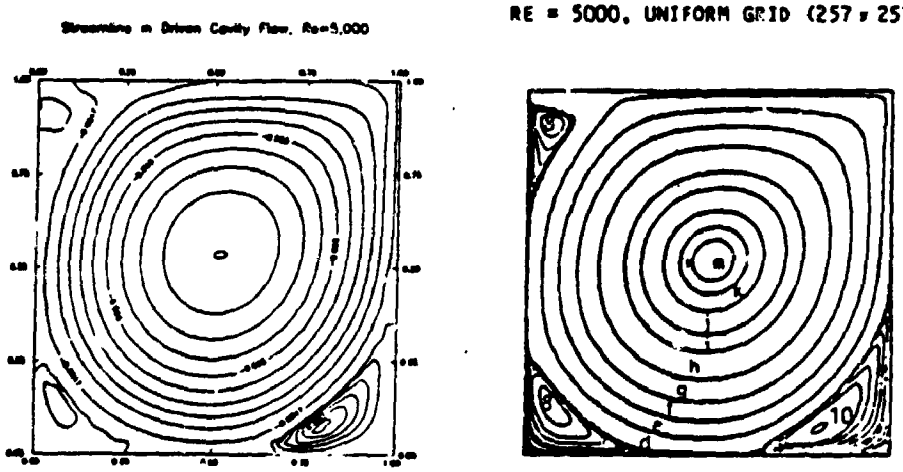
Figure 1.7 Stream function contours at Re = 100



Present Results

Results of Ghia *et al* (1982)

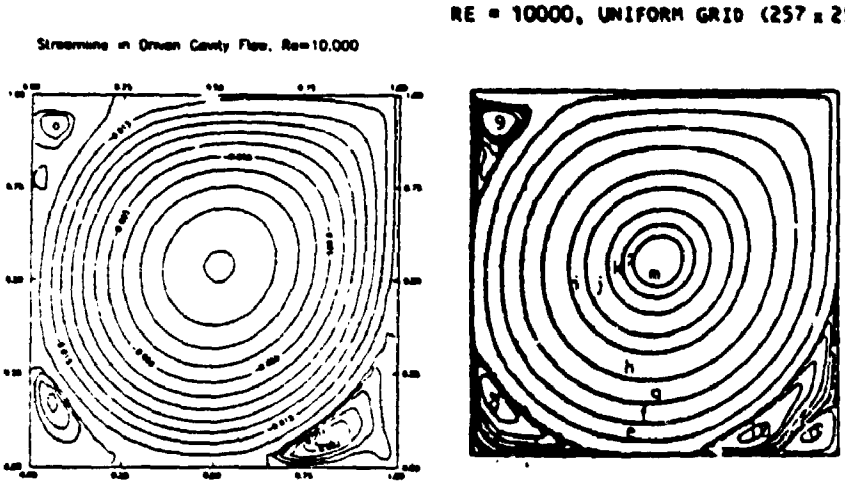
Figure 1.8 Stream function contours at Re = 1,000



Present Results

Results of Ghia *et al* (1982)

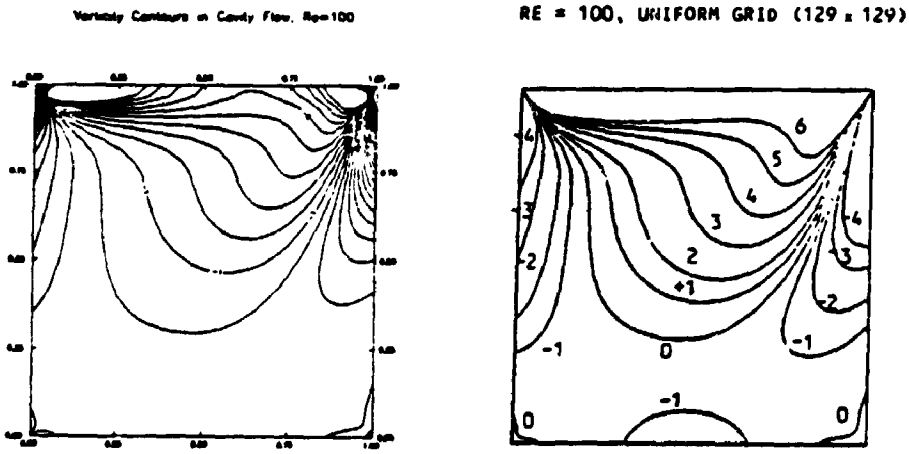
Figure 1.9 Stream function contours at Re = 5,000



Present Results

Results of Ghia *et al* (1982)

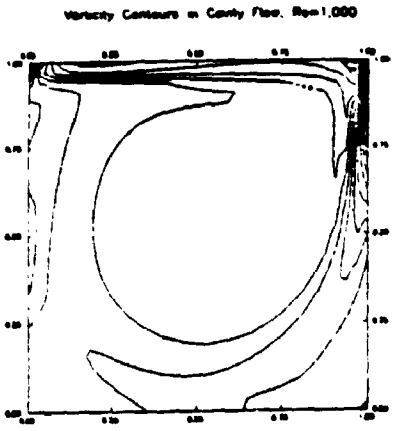
Figure 1.10 Stream function contours at Re = 10,000



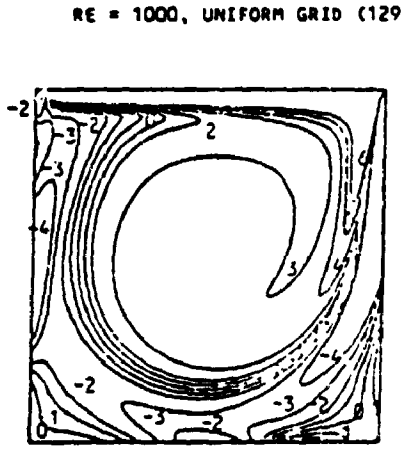
Present Results

Results of Ghia *et al* (1982)

Figure 1.11 Vorticity contours at Re = 100

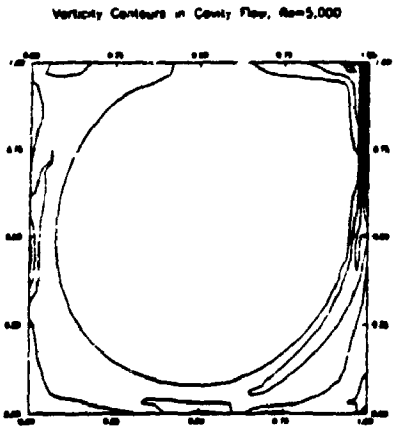


Present Results

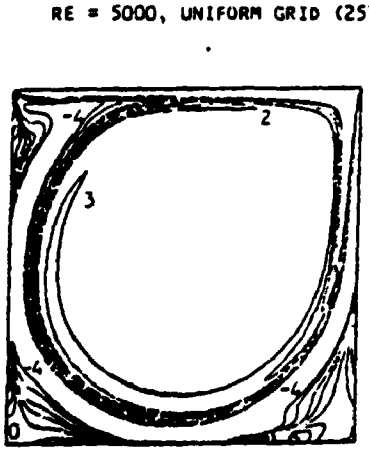


Results of Ghia *et al* (1982)

Figure 1.12 Vorticity contours at Re = 1,000

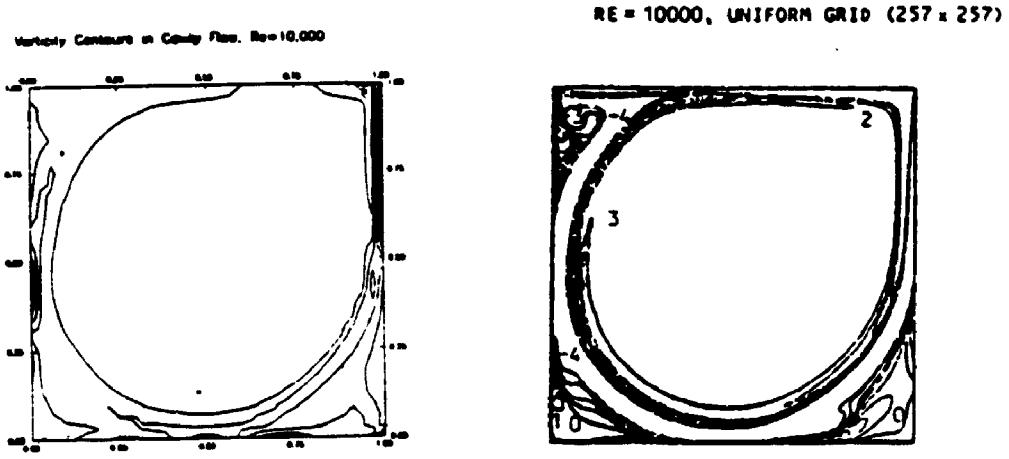


Present Results



Results of Ghia *et al* (1982)

Figure 1.13 Vorticity contours at Re = 5,000



Present Results

Results of Ghia *et al* (1982)

Figure 1.14 Vorticity contours at Re = 10,000

presented together with results of Ghia *et al* (1982) in **Figure 1.7** through **Figure 1.14**. The plots obtained here used the results of Zero Perturbation Method in primitive variable formulation while Ghia *et al* used stream function-vorticity formulation. There is very little good results available in primitive variable formulation in the literature, especially for Reynolds number up to 5,000.

As we may see from these presented figures (i.e., **Figure 1.7** through **Figure 1.14**), the motion of top wall in the driven cavity problem sets up a complex vortex structure. At lower Reynolds numbers, the flow consists of a primary vortex and two secondary vortices at the bottom upstream and downstream. As the Reynolds number is increased beyond values of 1,000-2000, another vortex is formed at the upstream (left) corner of the cavity. The comparison of contours for both stream function and vorticity have shown the agreement with Ghia *et al* (1982). These flow patterns are also in good

agreement with the results of Schreiber and Keller (1984), Kim and Moin (1985). The minor vortices embedded in the bottom corners are not plotted here because the contour intervals were selected to be uniform, and the stream function in these location were of the order of 10^{-5}. The values of minimum streamline in the central vortex were provided in the previous section which are also in excellent agreement with those earlier results.

Figure 1.15 through **Figure 1.18** present the surface of vorticity distributions in the cavity. The plots were also obtained using the results of Zero Perturbation Method for Reynolds numbers 100, 1000, 5000, 10000. It is of interest that when the Reynolds number goes up to 10,000 the vorticity distribution in the core region becomes more and more flat. This indicates the higher Reynolds number of the flow, the stronger singularity along the side of the cavity. Therefore, the center part of the cavity will be a inviscid flow.

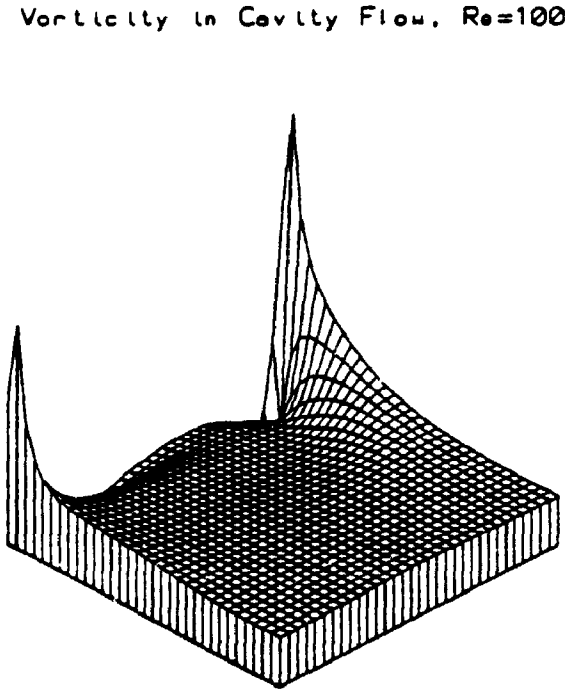


Figure 1.15 Vorticity surface at Re = 100

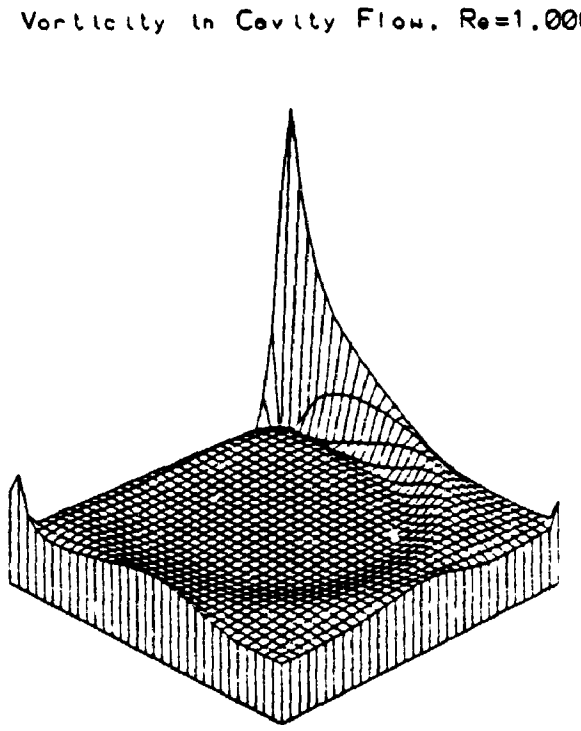


Figure 1.16 Vorticity surface at Re = 1,000

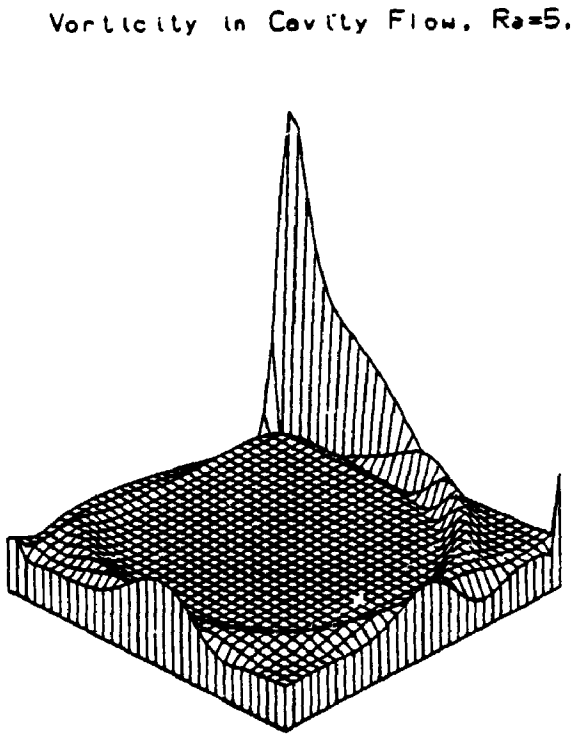


Figure 1.17 Vorticity surface at Re = 5,000

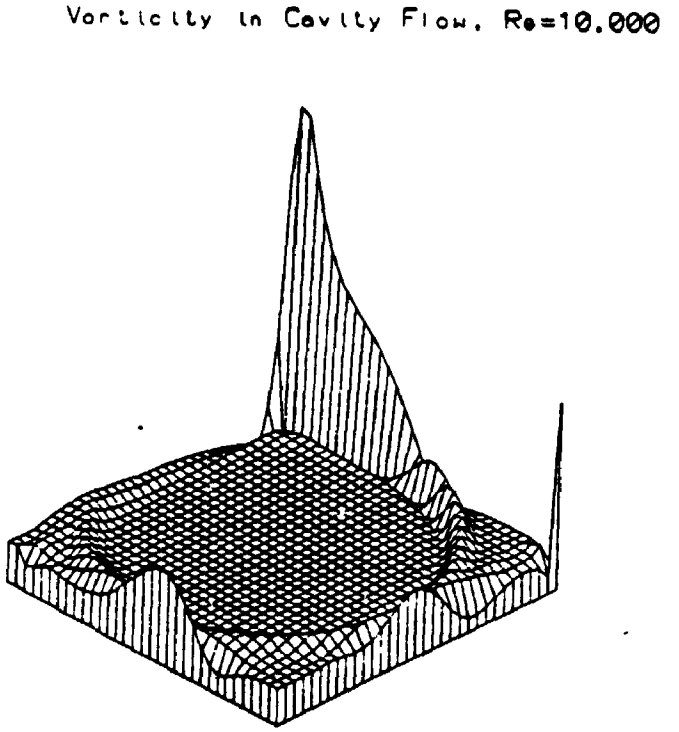


Figure 1.18 Vorticity surface at Re = 10,000

As a final comment for this section, the use of laminar flow equations up to 10,000 is predicated on the fact that there is no conclusive experiment evidence available for recirculating flow in two dimensional driven cavity. Therefore, no statement has been made that a transition to turbulence occurs at a lower Reynolds number. Clearly, it would be desirable for both experiments and flow stability analyses to be conducted to determine the critical Reynolds number for this geometry and flow situation. In any flow stability analysis, reliable solutions for the laminar flow equations will be required.

CHAPTER 5: SUMMARY

In this part of the thesis two new accurate finite-difference algorithms for the two dimensional, steady, incompressible Navier-Stokes equations are described. The numerical algorithms can be easily adapted on uniform grid for solving general flows. The first approach, which is designated as Zero Perturbation Method, uses the combination of the momentum and divergence equations at the boundary to provide the implicit pressure boundary conditions. The numerical solution procedure is discussed in primitive variable formulation. The second approach, which is designated as Computational Boundary Condition Method, utilizes a computational solution domain to avoid the problems of no explicit boundary conditions for pressure or vorticity in the Navier-Stokes equations. The pressure or vorticity boundary conditions are implicitly specified on the computational boundaries by the overspecified velocity or stream-function conditions. The latter approach seems more natural for the general flow problems. In this case, both stream function-vorticity and primitive variable formulations are discussed. In both finite-difference approaches, the overdetermined system resulting from the Navier-Stokes equations can be correctly reduced to a determined system.

All numerical experiments were performed on uniform grids by means of Zero Perturbation Method and Computational Boundary Condition method described in chapters three and four respectively. The accuracies of the two algorithms have been tested on both one and two dimensional model problems. Calculations of convergence rate are in excellent agreement with the theoretical prediction. Results for the driven cavity problem have found to be very accurate in comparisons with the available results

obtained by previous researchers. Although the numerical experiments are only carried on one and two dimensional simple geometry problems, the methods are suitable for solving problems with complex geometry, especially in three dimensional cases. With solutions for all the dependent variables obtained by the same order of accuracy, the methods have not only the theoretical significance, but wide scope of applications on practice as well.

PART II: NUMERICAL SIMULATION OF FREE/MOVING BOUNDARY PROBLEMS

CHAPTER 1: INTRODUCTION

1.1 The Background

Liquids in a low gravity environment freely float rather than staying in a container. Dynamics of liquids is dominated by free surface effects. It is the understanding of these effects that becomes crucial for our ability to predict the behaviour of liquids in the absence of gravity. Experimental investigation of the free surface effects is difficult in a terrestrial environment since they are masked by much stronger gravitational forces. One must therefore rely either on microgravity flight experiments or on theoretical modelling coupled with numerical simulation. There are only limited opportunities to use the former approach. The latter approach is more economical. However, the success of latter approach hinges on our capabilities of developing reliable and accurate algorithms for the analysis of the so-called moving boundary problems for the Navier-Stokes equations. The term "moving boundary problem" denotes here a problem where the domain of interest has an unknown boundary which has to be determined as a part of the solution procedure. In the numerical simulation the understanding of such physical process has to rely on algorithms capable of handling moving boundary problems for the Navier-Stokes equations. The two major difficulties involved in the simulation are: i) accurate tracking of curvature of an interface undergoing large deformations, and ii) diagnosing of the initiation of the break-up of an interface, i.e. break-up of a liquid drop. A successful resolution of these difficulties requires an algorithm capable of accurate determination of pressure and velocities along the deformed interface.

Batchelor (1967) and Davies and Rideal (1963) described the physical conditions of the moving boundary, which is a boundary between two immiscible fluids. When the surface tension at the interface is considered to be uniform, there are three interfacial boundary conditions: the kinematic condition, the tangential and normal stress conditions. The kinematic condition $\mathbf{V_A} \bullet \mathbf{n} = \mathbf{V_B} \bullet \mathbf{n}$, where $\mathbf{V_A}$ and $\mathbf{V_B}$ denote the velocity of two immiscible fluids A and B, expresses the fact that the interface is a material surface. Expression $\mathbf{V_A} \bullet \mathbf{t} = \mathbf{V_B} \bullet \mathbf{t}$ simply states that the tangential velocity across an interface is continuous; $\mathbf{t}$ is a unit vector tangent to the interface. The force balance at the interface is described by $\mathbf{n} \bullet (S_A - S_B) = 2\kappa\sigma\mathbf{n}$. Here, S_A and S_B are stress tensors of fluids A and B respectively, κ is the mean surface curvature and σ is the surface tension.

One of the dominant factors affecting dynamics of liquids in a low gravity environment is surface tension and suitability of any algorithm for simulation of motion of liquids under such conditions could be judged based on its success in accounting accurately for the surface tension effects. The variations of surface tension are usually not taken into consideration in the simulation due to the fact that the mechanical properties of an interface are not well understood in this kind of situation.

The contact line could be described as the interface comes into contact with a solid lines. There are several choices of contact line conditions in the physical modelling (Dussan 1979). Motion of fluid in the neighbourhood of a contact line usually leads to a discontinuity at the contact point. The contact conditions could be described as a fixed contact line condition which is suitable in the case of static contact

lines; a fixed contact angle condition which is suitable in the case of dynamic contact lines; or a mixed condition of two previous cases.

The governing equations of flow field are the Navier-Stokes equations supplemented by the standard no-slip boundary conditions and no-penetration conditions, by the inflow and outflow conditions when the flow through the domain of interest occurs, and by the interfacial conditions described earlier. The system of equations of motion is closed sufficiently by the number of interfacial boundary conditions to determine the location of the interface. The available numerical solutions have confirmed such problems are well posed. However, the mathematical properties of these initial boundary value problems, i.e., the condition of existence, uniqueness are still not clear. Several numerical methods involving the use of continuum models as well as discrete particle simulation have fulfilled their success. Detailed review of different numerical methods can be found in Floryan and Rasmussen (1989). Yeung (1982) reviewed the algorithms developed for the analysis of surface waves while Hyman (1984) and Laskey *et al* (1987) described methods for numerical tracking interfaces. Other successful methods dealing with the moving boundary problems are by Crank (1984), Bulgarelli *et al* (1984), and Harlow (1969).

1.2 The Methods of Interface Tracking

The major difficulty in the numerical simulation of moving boundary problems is the determination of interface. Therefore, the moving surface usually requires special attention. Four major techniques for interface tracking are described here: surface-tracking methods, moving-grid methods, volume-tracking methods, and gradient

methods.

The surface-tracking methods have been used by Glimm *et al* (1983) and Chren *et al* (1986). It represents an interface as a connected series of interpolated curve through points on the interface. At each time step, the points are saved in an array along with the information about the sequence in which they are connected. These points are advected with the flow field. In the simplest forms of surface-tracking methods for two dimensions, the points are saved as a sequence of height above a given reference line. The approach fails if the interpolated curve is multivalued or does not extend all the way across the region. However, this problem may be avoided if the points follow a parametric representation. This formulation is more complex, but it can represent fine detail in the interface if enough points are used. The feature of surface-tracking methods is that they can resolve features of the interface that are smaller than the cell spacing of the macroscopic Eulerian grid on which the curves are overlaid. There is naturally a price paid for storing this additional information. The time step for the entire calculation can be limited by the amount of movement the interface can go during each time step. There are some difficulties in this method, such as the problem of handling merging interfaces or joining a part of an interface to itself; and the points can accumulate in one segment of the interface leaving other segments without enough resolution. The first one requires re-ordering the interface points, which could require significant computational efforts. For the most accuracy, the best way is to limit the largest distance between neighboring points to be something less than minimum size of the local computation grid (Hirt and Nichols 1981) . Interface areas typically increase continually in complex flows. Thus it is necessary to add points along the interface

automatically. Conversely, points should be deleted where there are too many. When points must be added or deleted, the best way to interpolated new points, to represent and to manipulate contours with changing lengths are major issues.

The grid is defined in such a way that the interface is always located on cell boundaries. We call this the moving-grid method. Maintaining a cell boundary between different fluids controls numerical smearing that can occur at the interface as the fluid is transported. The interface is then a well defined continuous curve because it coincides with cell boundaries. There several approaches to actually implementing this idea. One is to maintain a grid of distorted quadrilaterals (Hyman and Larrouturou 1982). Another approach is to use a generalized orthogonal grid that fits the form of the interface.

One of the popular interface tracking methods is the volume-tracking method, which was first proposed by Harlow (1955). The method reconstructs the interface whenever it is needed rather than store representation of the interface in the surface-tracking approach. The earliest volume-tracking method used marker particles so that the density of particles in each cell indicates the density of the material. In the Marker-and-Cell or MAC method (Harlow and Welch 1965, Welch *et al.* 1966) the particles are tracers, marker particles with no mass.

We now consider some features of MAC method implemented by Amsden (1966). It uses as Eulerian grid in which velocity, internal energy, and total cell are defined at cell centers. In addition, the different fluids are represented by Lagrangian mass points, the marker particles, that move through the Eulerian grid. The marker particles each have a constant mass, a specific internal energy, and a recorded location in the Eulerian grid, and are moved with local velocity. The particle mass, momentum,

and external energy are transported from one cell to its neighbor when a marker particle crosses the cell boundary. Cells containing marker particles of both fluids contain the interface. Since the interface can be reconstructed locally at any time, the problems associated with interacting interfaces and large fluid distortions are eliminated. The method generalizes to any number of fluids.

Marker-particle methods cannot resolve details of the interface which are smaller than mesh size. It is expensive with respect to their requirements in computer time and memory. Particles may accumulate in portions of grid, leaving other portions not well resolved. Since mass, momentum, and energy are associated with each particle in marker particle method, adding and deleting particles is not very straightforward. Some fluctuations may not be acceptable if there are not enough marker particles and if variation of the marker particles is too large.

One of the features of the volume-tracking method is the use of fraction of a cell volume occupied by one of the materials as the marker for reconstructing the interface. If this fraction is zero for a given cell, the material does not occupy the cell and there is no interface in that cell. Conversely, if the fraction is one, the cell is completely occupied by the material and again there is no interface present. An interface is constructed only if the fractional marker is between zero and one.

In the simple line interface calculation method, each grid cell is partitioned by a horizontal or vertical line such that the volume of the partitioned part of the cell equals the fractional marker volume. The algorithm SLIC was first proposed by Noh and Woodward (1976), and improved by Chorin (1980) who added a corner interface to the straight horizontal and vertical lines and kept the same fractional cell volume as the

variable for locating the interface.

The VOF method described by Hirt and Nichols (1981) uses a straight line (with any slope) to represent the interface within a cell. A numerical estimate of the x-direction and y-direction derivatives of the volume of fluid occupying a cell are obtained for a given cell. Given the slope of the interface and side of the interface on which the fluid is located, then the position of the interface within the cell is set. The process is done for every cell with the value of its occupied volume between zero and one. Hirt and Nichols use values of the fractional volume of fluid averaged over several cells to calculate the derivatives. Their method depends on the ability to advect the volume fraction through the grid accurately without smearing from the numerical diffusion. Hirt and Nichols have described a donor-accepter method to insure that only the appropriate constituent fluid moves to or from a cell containing an interface. This helps to avoid averaging that results in numerical diffusion.

The gradient method (Laskey *et al* 1987) represents the interface as relatively continuous over several cells rather than to define the exact location of the interface within a cell. By keeping the resolution at the limit of that of the numerical convection algorithm, the amount of computer storage requirements and cost of interface tracking can be reduced. Laskey has applied his idea to flame fronts, although it may be useful for other types of interfaces. The method ensures that the right amount of reaction takes place in the vicinity of the gradient, as defined by the macroscopic grid. Also, it treats the effects of merging interface with relatively little difficulty. Adding other interface processes, and eventually ignoring weakened interfaces results naturally from the formulation. Because no additional variables are needed, computer memory

requirements are modest. The algorithm as implemented is fully vectorized. Finally, it is straightforward to extend the two dimensional formulation to three dimensions. The disadvantage of the gradient method is that the location of the reaction front is only approximately known in comparison with the volume-tracking method. Thus if it is necessary to track the curvature at the surface on scales comparable or smaller than the Eulerian grid spacing, another method should be used. The gradient method is not suited to simulate the dynamic droplet problem because the accurate curvature of a surface is essential for the calculation the surface-tension effects (Fyfe *et al* 1987).

1.3 The Finite Difference and Finite Element Methods

Finite difference method uses the direct substitution of the derivatives in the field equations by their finite difference approximation which result in the difference analogue of the original field equations. The application of this analogue to each of the mesh points in the flow field will result in a system of difference equations which can be solved to obtain the required numerical solution. The finite element method divides the solution domain into a finite number of subdomains or elements to solve the field equations. These elements may be one, two or three dimensional according to problem considered. The shape of element can be chosen either triangle, rectangle, quadrilateral, etc. The degrees of freedom of an element depend on the number of nodes in that element and the number of variables associated with each node. Once the type of element has been decided and the finite element mesh has been constructed, the unknown variables on each element can be approximated by continuous functions expressed in terms of nodal values of the unknown variables and their derivatives. This

also formulates a system of discrete equations and its numerical solutions can be obtained.

One of the successful finite difference methods for the steady, two dimensional free surface flows was discussed by Ryskin and Leal (1984) which is essentially based on the construction of boundary fitted orthogonal curvilinear coordinate system. The scheme consists of a Picard type iteration on the normal stress boundary condition, where for each approximation of the shape of the boundary a new boundary conforming coordinate system is constructed numerically. The field equations are solved by the ADI solution procedure using a stream function-vorticity formulation. The algorithm was used to solve buoyancy-driven motion of a gas bubble through quiescent liquid and to bubble deformation in an axisymmetric straining flow. Similar to the steady flow problems, Kang and Leal (1987) propose a fully implicit, backward time differencing procedure for the two dimensional, unsteady flows. The Picard solution procedure was kept, while the flow field and location of the interface approximately predicted from the information given by previous time step. The nonlinear algebraic equations describing the flow field at a new instant in time are also solved by the ADI solution procedure. The successful explicit approach using primitive variables was proposed by Miyata *et al* (1987). The kinematic condition was used to determine the new location of the interface which is defined in terms of a height function. In each iteration, the velocity and pressure field are determined by the field equations and the simplified interfacial conditions in which the viscous stress tensor is omitted. The calculation was made on the analysis of flow around a ship hull.

The numerical solutions of the viscous, steady flows with moving boundaries

are commonly obtained by using finite element method in Galerkin forms. The field equations are mostly expressed in terms of primitive variable approaches, and some are solved by penalty or collocation approaches. The excellent descriptions of the finite element procedures including some standard programs for the general flow problems are provided by Becker *et al* (1981) and Fletcher (1984). The finite element method using Galerkin formulation to solve die swell problems with a free surface for zero Reynolds number flow was studied by Nickell *et al* (1974). They used the kinematic condition to determine the free surface. Later, this approach was extended to the flow problems in consideration of surface tension as well as Reynolds number up to 50 (Reddy and Tanner 1978) which agreed with the experimental results.

Another Picard type iteration in finite elements is the use of normal stress boundary condition for the determination of the free surface. Orr and Scriven (1978) applied the Galerkin approach to the rimming flow problem. Silliman and Scriven (1980) used the boundary collocation coupled with Newton iteration in the treatment of normal stress condition for die swell problem. Lowndes (1980) utilized the solution of the Laplace-Young equation of the capillary dynamic hydrostatic in accounting of the dynamic effects to analyse the creeping flow in a capillary tube in absence of gravity.

The determination of the free surface in finite element methods may also be achieved by using the direct solutions in which the kinematic, tangential and normal stress conditions are solved simultaneously with the field equations. Ruschak (1980) proposed an effective method by considering the values of the parameters to be selected in such a way that the surface line coincides with the grid line. The more general approach was later given by Dupret (1982). Some comparisons of the solutions by

Picard and direct methods were discussed by Cuelier and Driessen (1986). For the unsteady flows, each finite element is a region in both space and time dimensions. Temporal discretization is obtained in a manner analogous to the spatial discretization. This results in a fully implicit procedure. By means of space-time finite element Galerkin approach, Ferderiksen and Watts (1981) used a computational procedure which consisted of predicting the flow field and location of the interface at the new time step from information obtained previously. Their solutions, however, only applied to the low Reynolds number flows.

1.4 The Pressure Problem in the Navier-Stokes Equations

There are many effective numerical methods in dealing with moving boundary problems with small to medium deformations. With a specific set of conditions the algorithms may work best but are not totally general purpose. There are no general criteria to assess the exact level of accuracy in these algorithms, and only experience indicates the capability of the accuracy achieved for the fixed boundary problems. For medium or large deformations, the accuracy of these methods still remains doubtful.

The inaccurate simulation of moving boundary problems may attribute to the fact that the numerical solution of the Navier-Stokes equations (in primitive variable formulations) itself has been plagued with confusion regarding the appropriate velocity and pressure conditions. It is now generally recognized that the staggered grid introduced by Harlow and Welch (1967) in Marker-and-Cell is not quite an accurate technique in solving the moving boundary problems because the normal stress

condition may not be used appropriately in the staggered grid. There are a large volume of papers that contain the discussion of possible pressure boundary conditions for the Navier-Stokes equations. However, the difficulty still exists in the numerical calcultion and the pressure problem remains unsolved.

Fortunately, the situation is not so hopeless as made above. There are still some successful numerical methods in the last decade to obtain the accurate pressure solutions. The influence matrix method by Kleiser and Schumann (1980) introduces a supplementary (linear) problem to determine the lacking boundary values for the pressure. Their method is quite efficient and has been employed to compute the three dimensional channel flow. The improvement of Marker-and-Cell made by Patankar and Spalding (1970) that an approximate equation of pressure correction is derived from the momentum equations to satisfy incompressible equations, instead of solving Poisson equation for pressure in Marker-and-Cell method. The projection method for the primitive variable formulation originally developed by Chorin (1967) sets up an auxiliary vector field V^* in which the momentum and incompressibility conditions are treated into two fractional steps. At the second step, the Neumann condition for pressure is made by projecting the vector field V^* onto its subspace with zero divergence and satisfying the appropriate boundary conditions. The improvement of Chorin's method has been made by Stephens, Bell, Solomon and Hackerman (1984) who developed the finite difference Galerkin method for the numerical solutions of the steady, incompressible Navier-Stokes equations. A Galerkin solution procedure is proposed by means of constructing a local basis for discretely divergence-free vector

field. Their method is quite appreciated by many researchers for the solutions of primitive variable formulation.

1.5 Outline of Part II

Apart from the pressure problem in the numerical solution of the Navier-Stokes equations, the difficulties in the simulation of a dynamic capillary surface include: 1) accurate tracking of the shape and curvature of the interface, 2) accurate treatment of singularities at the contact points, 3) decision algorithms determining existence or non-existence of the dynamic capillary surface, and 4) decision algorithms permitting separation of the physical and purely numerical interface instabilities. There are several existing algorithms for this class of problems and they could be broadly classified as either Eulerian or Lagrangian. See Floryan and Rasmussen (1990) for a review. While these algorithms in most instances reproduce the correct physics of flow, their accuracy is uncertain. The main goal of this work is, therefore to provide an accurate and reliable algorithm for the simulation of moving capillary surfaces by applying the Zero Perturbation Method discussed and tested in Part I in which the pressure problem has been completely eliminated. In chapter two we describe a model problem that is used to test our algorithm. In chapter three we describe a solution procedure valid in the limit of large surface tension. An algorithms used to solve field equations and some results of our tests are presented in chapter four. Finally a summary is given in chapter five.

CHAPTER 2: PROBLEM FORMULATION

We consider a two-dimensional steady flow of an incompressible Newtonian liquid in a cavity of length L and heights H_L, H_R, as shown in **Figure 2.1.** The liquid has its density ρ and kinematic viscosity $\upsilon = \mu/\rho$; μ is the dynamic viscosity. The upper surface is free surface associated with a surface tension σ, and is bounded by a passive fluid of negligible density and viscosity.

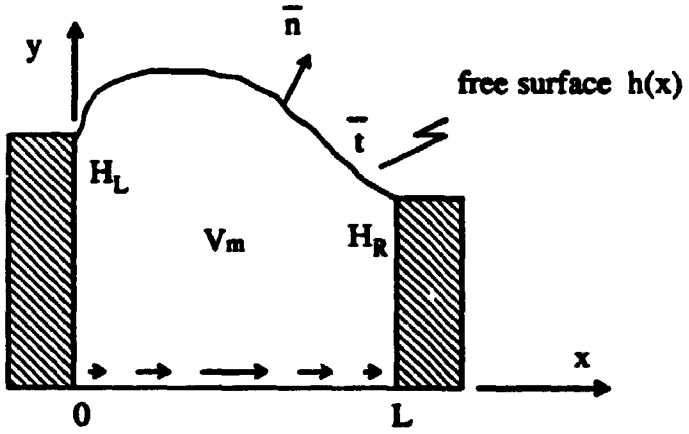


Figure 2.1. Schematic Diagram of the Model Problem

The x and y coordinate axes are horizontal and vertical, respectively, with origin located at the low left of the cavity. Location of the surface is described by $y=h(x)$. The corresponding components of the velocity are u and v, while the flow is driven by specifying the tangential velocity component at the bottom of the cavity. In the absence of gravity force, the motion of the liquid is governed by the equations

$$\mathbf{V}\bullet\nabla\mathbf{V} = -\rho^{-1}\nabla p + \upsilon\nabla^2\mathbf{V}, \tag{II-1}$$

$$\nabla\bullet\mathbf{V} = 0, \tag{II-2}$$

where $\mathbf{V} = u\,\mathbf{i} + v\,\mathbf{j}$ is the velocity vector and p is the pressure of the liquid. These equations are subject to the following boundary conditions:

$$x = 0, x = L: \quad u = v = 0 \tag{II-3}$$

$$y = 0: \quad u = 16U[(1-x)x]^2 \tag{II-4}$$

$$v = 0 \tag{II-5}$$

$$y = h(x): \quad v = h_x u \tag{II-6}$$

$$S_{ij}n_j n_i = \sigma\kappa \tag{II-7}$$

$$S_{ij}n_j t_i = 0. \tag{II-8}$$

In the above, h_x is the first derivative of surface deformation h(x), κ denotes the curvature of the interface and, n_i, t_i stand for unit normal and tangential vector respectively.

In Eq.(II-4) the driven velocity, which has its maximum value at the geometric center, is chosen to smooth out the corner singularities. Eq. (II-6) is the kinematic condition at the interface, i.e., a liquid-gas interface, which expresses the fact that the velocity normal to the interfacial streamline should be equal to zero. Eqs.(II-7)-(II-8) are the free surface stress boundary conditions, which state the principles on the basis that: i) stress S_{ij} tangential to the free surface must be vanish; ii) stress S_{ij} normal to the surface must exactly balance any externally applied normal stress. The first principle

implies that the external gas be incapable of exerting a shear stress on the interface. The second principle allows for the driven surface motions to be affected by the surface tension, and by external pressure whose inertial contribution to the dynamics is negligible. It implies that the surface tension, pressure variations in the gas are the major effects of its dynamics.

The stress conditions resulting from the free surface boundary are complicated. To illustrate these conditions explicitly, Let n_x and n_y be components of a unit outward vector n_i, normal to the surface. Then,

$$n_x = -\frac{h_x}{\sqrt{1+h_x^2}}, \qquad n_y = \frac{1}{\sqrt{1+h_x^2}}. \tag{II-9}$$

If θ represents the angle of the unit tangential vector to x-axis, we have $\tan\theta = -n_x/n_y$. In addition, let p_0 be the externally applied pressure. Then, within the fluid at its interface, the tangential stress condition (II-8) becomes

$$\sin 2\theta\,(v_y - u_x) + \cos 2\theta\,(u_y + v_x) = 0, \tag{II-10}$$

while that for the normal stress can be written

$$\sigma\kappa = -(p - p_0) + 2\mu\,[\,\sin^2\theta\,u_x - \cos\theta\sin\theta\,(u_y + v_x) + \cos^2\theta\,v_y\,]\,. \tag{II-11}$$

It should be mentioned that the normal stress condition (II-11) that includes the surface tension into the free surface boundary represents a class of problems, which are difficult

to deal with numerically. In this thesis an effective way to incorporate this effect into the numerical calculation has been found and is discussed later.

From the fundamental fluid mechanics theory, we may find that the stresses will become infinite in the neighborhood of free surface that contacts to the end walls. In the present case, we specify the type of contact as the liquid sticking to the sharp edge at the end walls, i.e.,

$$h(0) = H_L\,, \qquad h(L) = H_R\,. \tag{II-12}$$

which corresponds to the case of fixed contact points at the walls.

Finally, to close the problem, it is clear that the liquid must also satisfy the mass conservation constraint, i.e., its volume must remain constant.

$$\int_0^L h\,(x)\,dx \;=\; V_m. \tag{II-13}$$

CHAPTER 3: NUMERICAL SIMULATION OF DYNAMIC CAPILLARY SURFACE

In this chapter we provide the numerical solution procedure based on asymptotic series expansion. The general solution procedure described in this part of thesis is the zero Reynolds number flow with large surface tension.

Before giving the details of the asymptotic solution procedure, we summarize the algorithm as follows (Some notations will be given in Section 3.2):

1) select the volume V in the cavity and velocity profile at the bottom.

2) determine the shape of the static free surface $h_0(x)$ by solving the following coupled equations numerically (or analytically):

$$-p_s = \frac{h_{0xx}(x)}{(1+h_{0x}^2)^{3/2}}, \tag{II-14}$$

$$\int_0^1 h_0(x)\,dx = V\,. \tag{II-15}$$

3) find the first order approximation of the flow by solving the Stokes equations

$$u_{0x} + v_{0y} = 0, \tag{II-16}$$

$$u_{0xx} + u_{0yy} = p_{0x}, \tag{II-17}$$

$$v_{0xx} + v_{0yy} = p_{0y}, \tag{II-18}$$

with the velocity boundary conditions at the solid walls, and the interfacial conditions given earlier. The interface conditions include the leading terms of kinematic and tangential conditions:

$$v_0 = h_{0x} u_0 \ , \qquad \text{at} \quad y = h_0(x), \quad \text{(II-19)}$$

$$2h_{0x}(v_{0y} - u_{0x}) + (1 - h_{0x}^2)(v_{0x} + u_{0y}) = 0, \quad \text{at} \quad y = h_0(x) \ . \quad \text{(II-20)}$$

This system of equations can be solved by introducing the transformation $y = z \cdot h_0(x)$ which is similar to Picard solution procedure of using the normal stress condition to determine the location of the interface.

4) find the free surface correction $h_1(x)$ by solving

$$\frac{h_{1xx}}{(1 + h_{0x}^2)^{3/2}} - \frac{3h_{0x}h_{0xx}h_{1x}}{(1 + h_{0x}^2)^{5/2}} =$$

$$-p_0 + \frac{2}{(1 + h_{0x}^2)} \{ (v_{0y} - h_{0x}u_{0y}) + h_{0x}(-v_{0x} + h_{0x}u_{0x}) \} \qquad \text{(II-21)}$$

in addition with

$$\int_0^1 h_1(x)\, dx = 0; \qquad \text{(II-22)}$$

5) determine asymptotically the final shape of free surface as

$$h = h_0(x) + h_1(x)\varepsilon. \tag{II-23}$$

In the above, u_0, v_0, p_0, h_0 are the first order of the series expansions; p_s is the static pressure due to the curvature; $h_1(x)$ is the second-order term of the moving surface; and ε is defined as the capillary number.

3.1 Mathematical Formulation of the Model Problem

We scale the model problem by introducing the following prime quantities:

$$\begin{aligned} x = Lx', y = Ly', h = Lh' \\ u = Uu', v = Uv', p = \frac{\mu U}{L}p' \end{aligned} \tag{II-24}$$

We notice that the pressure scaling is given as lubricant pressure scaling for the low Reynolds numbe- flow rather than the common use of dynamic pressure scaling. With these scales, the dimensionless equations (with prime dropped) can be written in the form:

$$u_x + v_y = 0 \tag{II-25}$$

$$Re(uu_x + vu_y) = -p_x + (u_{xx} + u_{yy}) \tag{II-26}$$

$$Re(uv_x + vv_y) = -p_y + (v_{xx} + v_{yy}). \tag{II-27}$$

The Reynolds number has the standard definition

$$Re = \frac{UL}{\upsilon}. \tag{II-28}$$

The boundary conditions (II-3)-(II-8) transform to

$$x = 0, x = 1: \quad u = v = 0 \tag{II-29}$$

$$y = 0: \quad u = 16[(1-x)x]^2 \tag{II-30}$$

$$v = 0 \tag{II-31}$$

$$y = h(x): \quad v = h_x u \tag{II-32}$$

$$-p + \frac{2}{(1+h_x^2)}\{(v_y - h_x u_y) + h_x(-v_x + h_x u_x)\} = -\frac{\varepsilon h_{xx}}{(1+h_x^2)^{3/2}} \tag{II-33}$$

$$2h_x(v_y - u_x) + (1-h_x^2)(v_x + u_y) = 0. \tag{II-34}$$

In Eq.(II-33), ε is the surface tension number, given by

$$\varepsilon = \frac{\mu U}{\sigma}. \tag{II-35}$$

The surface tension number measures the degree of deformation of the free surface caused by the motion of the liquid; $\varepsilon \to 0$ implies a very strong surface tension and small deformation of the free surface.

The dimensionless form of the mass constraint condition (II-13) is as follows:

$$\int_0^1 h(x)\,dx = V, \qquad \text{where} \quad V = V_m/L^2, \tag{II-36}$$

and the contact line conditions are assumed the following form:

$$h(0) = h(1) = 1, \quad \text{where} \quad H_L = H_R = L. \tag{II-37}$$

We will determine the solution to the flow problem described by Eqs.(II-25)-(II-27) subject to boundary conditions (II-29)-(II-34) and constraint conditions (II-36)-(II-37).

In the present chapter we will investigate only the Stokes approximation $Re \to 0$. The next section deals with the solution in the asymptotic limit of the small surface tension number $\varepsilon \to 0$. The deformation of the interface decouples from the flow equations then, and the flow reduces in principle to the flow in a cavity with upper surface rigid. The deformation of the interface is determined as a higher order correction in the asymptotic expansion.

3.2 Very Large Surface Tension

We seek a solution of the problem (II-25)-(II-37) in the asymptotic limit of small surface tension number $\varepsilon \to 0$. We note that the capillary effects are dominant in this limit and therefore, these effects determine the form of the interface at the leading order of approximation. The dynamic effects associated with the motion of the liquid produce deformation of the interface. However this deformation is of the lower order of magnitude, where ratio of dynamic and capillary forces, i.e., the surface tension number ε, is used to define the appropriate ordering. Accordingly, we write the expansions (Rybicki and Floryan 1987)

$$u = u_0 + u_1\varepsilon + O(\varepsilon^2) \tag{II-38}$$

$$v = v_0 + v_1\varepsilon + O(\varepsilon^2) \tag{II-39}$$

$$p = \varepsilon^{-1}p_s + p_0 + p_1\varepsilon + O(\varepsilon^2) \tag{II-40}$$

$$h = h_0 + h_1\varepsilon + O(\varepsilon^2). \tag{II-41}$$

The first term in the pressure expansion corresponding to the capillary pressure due to the surface tension and curvature of the interface. The magnitude results from the rescaling of this component with the capillary pressure scale σ/L.

Substituting expansions (II-38)-(II-41) into the field equations (II-25)-(II-27), boundary conditions (II-29)-(II-34) together with the constraint (II-36) and contact condition (II-37), we find:

the problem of order T^{-1}:

$$p_{sx} = 0 \tag{II-42}$$

$$p_{sy} = 0 \tag{II-43}$$

with the boundary condition

$$-p_s = \frac{h_{0xx}(x)}{(1+h_{0x}^2)^{3/2}}; \tag{II-44}$$

the problem of order T^0:

$$u_{0x} + v_{0y} = 0 \tag{II-45}$$

$$Re\,(u_0u_{0x} + v_0u_{0y}) = -p_{0x} + (u_{0xx} + u_{0yy}) \tag{II-46}$$

$$Re\,(u_0 v_{0x} + v_0 v_{0y}) = -p_{0y} + (v_{0xx} + v_{0yy}) \tag{II-47}$$

with the boundary conditions:

$$x = 0, x = 1: \quad u_0 = v_0 = 0 \tag{II-48}$$

$$y = 0: \quad u_0 = 16[(1-x)x]^2 \tag{II-49}$$

$$v_0 = 0 \tag{II-50}$$

$$y = h_0(x): \quad v_0 = h_{0x} u_0 \tag{II-51}$$

$$2h_{0x}(v_{0y} - u_{0x}) + (1 - h_{0x}^2)(v_{0x} + u_{0y}) = 0 \tag{II-52}$$

$$\frac{h_{1xx}}{(1+h_{0x}^2)^{3/2}} - \frac{3h_{0x}h_{0xx}h_{1x}}{(1+h_{0x}^2)^{5/2}} =$$

$$-p_0 + \frac{2}{(1+h_{0x}^2)}\{(v_{0y} - h_{0x}u_{0y}) + h_{0x}(-v_{0x} + h_{0x}u_{0x})\} \tag{II-53}$$

the volume constraint condition

$$\int_0^1 h_0(x)\,dx = V \tag{II-54}$$

and the contact conditions

$$h_0(0) = h_0(1) = 1. \tag{II-55}$$

We shall need the following constraint and contact conditions of order T^1 to complete the system of equations:

$$\int_0^1 h_1(x)\,dx = 0 \tag{II-56}$$

$$h_1(0) = h_1(1) = 0. \tag{II-57}$$

The solution of the leading order approximation is defined as a meniscus problem. This approximation is described by Eqs.(II-42)-(II-44). We find from Eqs.(II-42)-(II-43) that the capillary pressure component is constant. We identify Eq.(II-44) as the Young-Laplace equation written in a form suitable for two dimensional configurations. Only interfaces of the constant curvature satisfy this equation. The function $h_0(x)$, therefore, describes an element of an arc or straight line. The particular forms of the interface depend on the volume of the liquid and contact conditions, specified by Eqs.(II-54)-(II-55). Eqs.(II-42)-(II-43) in addition with Eqs.(II-54)-(II-55) completely determine the initial shape of the interface $h_0(x)$, therefore. With the shape of the interface $h_0(x)$ determined, we proceed to the determination of the flow inside the cavity.

The motion of the liquid is described by Eqs.(II-45)-(II-47) subject to boundary conditions (II-48)-(II-52). This is a closed system and we shall solve it numerically. Its boundary conditions do not involve pressure and the flow structure. Therefore, the motion can be determined with accuracy up to a constant in the pressure field. The detailed description of the solution method will be given in next section.

We may notice that Eq.(II-53) is not required for the solution of the flow

problem. This equation involves pressure p_0, deformation of the interface $h_1(x)$ and other quantities that have been determined in the meniscus problem and the flow problem. We utilize this equation for the simultaneous determination of the deformation of the interface and the free constant in the pressure field.

We write Eq.(II-53) as:

$$h_{1xx} + G(x)\, h_{1x} = H(x, p_0) \ , \tag{II-58}$$

where $G(x) = -3h_{0x}h_{0xx}(1+h_{0x}^2)^{-1}$ and $H(x,p_0) = -p_0(1+h_{0x}^2)^{3/2}$ $+ (1+h_{0x}^2)^{1/2}\{(v_{0y} - h_{0x}u_{0y}) + h_{0x}(-v_{0x} + h_{0x}u_{0x})\}$. This is a second-order, linear differential equation for $h_1(x)$, involving an undetermined constant in the pressure p_0. We have a total of three arbitrary constants in the general solution and the three required additional conditions are the volumatic constraint condition (II-56) and contact conditions (II-57). Eq.(II-58) forms a boundary value problem subject to the constraint (II-56).

3.3 Numerical Solution of Cavity Flow with Upper Surface Rigid (Small Reynolds Number Limit)

The governing equations (II-45)-(II-47) and the boundary conditions (II-48)-(II-52) for the cavity flow with upper surface rigid have been discussed in Section 3.2. The present section is devoted to the detailed analysis of the flow of motion in the Stokes limit (

$Re \to 0$). This case corresponding to the strong capillary effects, i.e., large surface tension, with the flow of motion consequently being relatively small. **Figure 2.2** shows the geometry, boundary conditions for the cavity flow where the volume V is a given condition in the numerical calculation. In the following, we drop the subscript of leading approximation term for the purpose of conciseness.

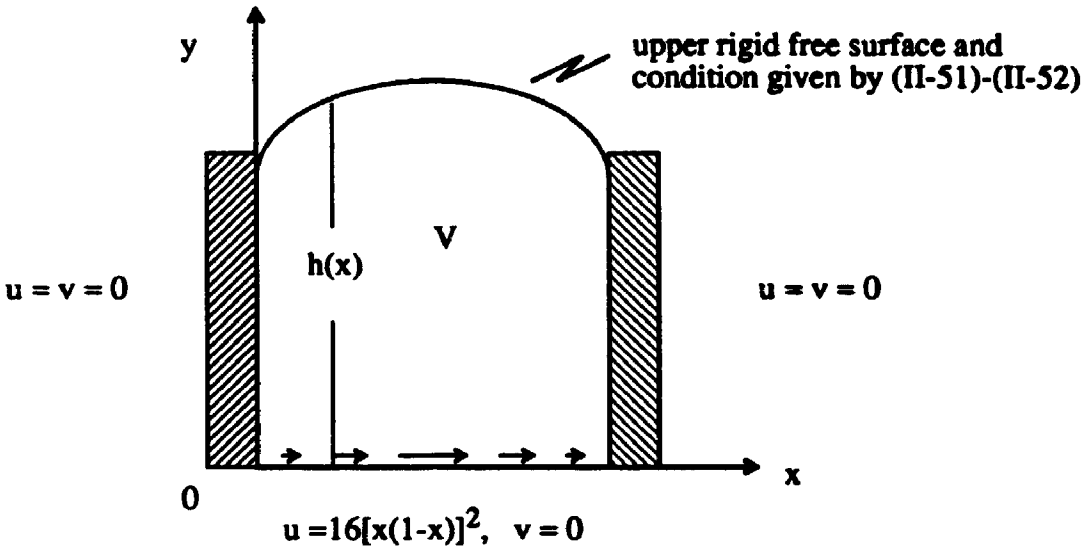


Figure 2.2. Geometry, boundary condition for the cavity flow

We adopt the mapping method to transform the irregularly shaped flow domain onto a regularly shaped computational domain. The mapping function $h(x)$ appears explicitly as a known function determined by the meniscus problem. The transformation is considered as

$$x = x, \qquad y = h(x)\,z, \tag{II-59}$$

in which transforms the domain $0 < x < 1, 0 < y < h(x)$ onto the domain

$0 < x < 1, 0 < z < 1$. The transformation facilitates application of the standard finite difference discretization procedure. The governing equations (II-45)-(II-47), therefore, lead to the form($Re \to 0$):

$$hu_x - zh_x u_z + v_z = 0 \tag{II-60}$$

$$p_x h^2 - zh_x h p_z = h^2 u_{xx} + (z^2 h^2 + 1)\, u_{zz} - 2zh_x h u_{xz} + z\,(2h_x^2 - h_x h)\, u_z \tag{II-61}$$

$$p_z h = h^2 v_{xx} + (z^2 h^2 + 1)\, v_{zz} - 2zh_x h v_{xz} + z\,(2h_x^2 - h_x h)\, v_z \tag{II-62}$$

where the convective terms have been omitted for the flow motion in the Stokes limit ($Re \to 0$). The boundary conditions for system are now given as

$$x = 0, x = 1: \quad u = v = 0 \tag{II-63}$$

$$z = 0: \quad u = 16\,[\,(1-x)x\,]^2 \tag{II-64}$$

$$v = 0 \tag{II-65}$$

$$z = 1: \quad v = h_x u \tag{II-66}$$

$$2h_x\,(v_z - hu_x + zh_x u_z) + (1 - h_x^2)\,(hv_x - zh_x v_z + u_z) = 0. \tag{II-67}$$

The above system of equations is to be solved for field variables u,v, and p.

After the transformation, right hand side of the normal stress condition (II-58) $H(x, p)$ which is used to determine the deformation of surface then becomes

$$H(x,p) = -p + \frac{2}{h(1+h_x^2)}\left\{ (v_z - h_x u_z) + h_x(-hv_x + zh_x v_z + h_x(hu_x - zh_x u_z)) \right\}$$

(II-68)

The new equation (II-68) will replace H(x,p) in (II-58) to solve for the second-order differential equation for the correction of surface deformation $h_1(x)$.

The reliability of the asymptotic approach relies on the accuracy of numerical solution of the field equations (II-60)-(II-62). The main issue involves accurate determination of pressure at the boundaries. Here, we apply artificial compressibility method with modifications necessary to strictly enforce the incompressibility conditions. The principle of artificial compressibility method is to consider the steady solution as the limit when t $\rightarrow \infty$of the solution of unsteady equations associated with a perturbed incompressibility equation. If the disturbance factor is small enough, the approximate solution of the Navier-Stokes equations would be made. The effectiveness of this method depends on the treatment of boundary conditions. We propose a modification of the original artificial compressibility method that involves the use of zero-perturbed momentum equations and incompressibility condition at the boundaries, which leads to an implicit pressure boundary conditions for the system. According to discussion in Part I, the algorithm has been confirmed for obtaining accurate pressure solution.

The field equations in forms of artificial compressibility method are

$$u_t + (h^2 u_{xx} + (z^2 h^2 + 1) u_{zz} - 2z h_x h u_{xz} + z(2h_x^2 - h_x h) u_z) - (p_x h^2 - z h_x h p_z) = 0 \qquad \text{(II-69)}$$

$$v_t + (h^2 v_{xx} + (z^2 h^2 + 1) v_{zz} - 2z h_x h v_{xz} + z(2h_x^2 - h_x h) v_z) - p_z h = 0 \qquad \text{(II-70)}$$

$$p_t + c^2 (hu_x - zh_x u_z + v_z) = 0. \qquad \text{(II-71)}$$

The above equations have a physical meaning only when the state $\frac{\partial}{\partial t} = 0$ is reached. The solution procedure is usually referred as to a pseudo-unsteady method because the time involved has no physical meaning. The velocity boundary conditions for the system are provided by (II-63)-(II-67). The velocity conditions at the top (z=1) could be solved implicitly based on the velocity values from the previous iteration, i.e, $v^{n+1} = h_x u$. Following Part I of the Zero Perturbation Method, the implicit Dirichlet pressure boundary condition from iteration to iteration are given as

$$p^{n+1} = p - \Delta t c^2 (hu_x - zh_x u_z + v_z)$$

$$+w_1[(h^2 u_{xx} + (z^2 h^2 + 1) u_{zz} - 2zh_x h u_{xz} + z (2h_x^2 - h_x h) u_z) - (p_x h^2 - zh_x h p_z)]$$

$$+w_2[(h^2 v_{xx} + (z^2 h^2 + 1) v_{zz} - 2zh_x h v_{xz} + z (2h_x^2 - h_x h) v_z) - p_z h]$$

(II-72)

at each boundary of domain $0 < x < 1, 0 < z < 1$, where w_1, w_2 are the parameters chosen as -1 or 1 for the stability purpose. In the above, the superscripts (n) for all the variables are omitted for the sake of simplicity.

We discretize the field equations (II-69)-(II-71) and the boundary conditions (II-66)-(II-67) and (II-72) on an uniform grid of constant step size using the standard second order finite difference approximations. The formulae for an interior point are

$$f'_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{II-73}$$

$$f''_i = \frac{f_{i-1} - 2f_i + f_{i+1}}{\Delta x^2} + O(\Delta x^2) \tag{II-74}$$

where f_i, f'_i, f''_i denote a function and its first and second derivatives, respectively, and subscripts i, $i-1$, $i+1$ denote values of the appropriate quantities at a grid point i and its two neighbours i -1 and i+1, respectively. At the boundary, we employ the one-side derivative approximations, i.e.,

$$f'_i = \frac{4f_{i+1} + 3f_i - f_{i+2}}{2\Delta x} + O(\Delta x^2) \tag{II-75}$$

$$f''_i = \frac{4f_{i+2} + 2f_i - 5f_{i+1} - f_{i+3}}{\Delta x^2} + O(\Delta x^2) \tag{II-76}$$

where i may correspond to a boundary point. Part of the algorithm involves differentiation of a function. This is done using (II-73)-(II-74) at the interior points and (II-75)-(II-76) at the boundaries whenever it is necessary.

The discretized equations are solved by means of ADI solution procedure consisting of a sweep for u and v in the x-direction followed by a similar sweep in the z-direction, which is then followed by updating the pressure and its derivatives using the implicit form of pressure Dirichlet condition (II-72). The x-sweep for u utilizes x-

momentum (II-69) combined with (II-73)-(II-74) at interior points, and with (II-75)-(II-76) and appropriate boundary conditions at the boundary points, to evaluate u, u_x, u_{xx} at a new time level $(n+\frac{1}{2})$. The x-sweep for v utilizes z-momentum (II-70) combined similarly with the (II-73)-(II-76) and the boundary conditions to evaluate v, v_x, v_{xx} at a new time level $(n+\frac{1}{2})$. The z-sweep for u utilizes x-momentum (II-69) combined with (II-73)-(II-76) and appropriate boundary conditions, to evaluate u, u_z, u_{zz} at a next time level $(n+1)$. The z-sweep for v is similar and leads to determination of v, v_z, v_{zz} at time level $(n+1)$. The pressure at time level $(n+1)$ is evaluated from (II-71) at the interior points and (II-72) at the boundaries.

The derivatives of p_x, p_z required for initiation of the next ADI step are evaluated through numerical differentiation of p^{n+1}. This is done with the help of the formulae (II-73) at the interior points and (II-75) at the boundary points. The parameter $\Delta t, c^2$ are selected to assure the stability of the numerical process of fast rate of convergence to a steady state. These may be referred to the results given in Part I. (pp.63 and pp.68)

CHAPTER 4: NUMERICAL RESULTS

4.1 Numerical Algorithm and Its Accuracy

Accurate evaluation of interface deformation requires knowledge of pressure and velocity at the interface (See Eq.(II-53)). The field equations (II-45)-(II-47) are supplemented by the Dirichlet-type boundary conditions for velocity while there are no boundary conditions for pressure. As discussed earlier, this lack of the pressure boundary conditions has been recognized as a major difficulty in numerical solution of the Navier-Stokes equations in terms of primitive variables. The common way to surmount this problem in finite difference method is to set up the staggered mesh in the discretization scheme, i.e., Marker-and-Cell mesh introduced by Harlow and We (1967), Projection method by Chorin (1968) and etc. These are among the many methods characterized by the appropriate treatment of the pressure conditions. However, these methods may not be applicable to the moving boundary flow problem since both velocity and pressure at the interface need to be determined.

We have introduced an algorithm that involves primitive variable formulation in which the equation of motion are solved directly for the unknown velocities and pressure. The pressure is determined using the concept of artificial compressibility with its modified implicit boundary condition. The method is designated as Zero Perturbation Method (see Part I). The principle of this method is to consider steady solution as the limit when $t \to \infty$ of the solution of unsteady equations associated with a perturbed divergence equation.

In our finite difference approach, the second-order discretization scheme is utilized incorporated with the ADI solution procedure for solving field equations. The parameters Δt, c^2 are selected to assure stability of the numerical process and fast rate of approach to a steady state. The appropriate values depend on the curvature of the interface. For a typical case of $\kappa = 0$, 21×21 grid, $\Delta t = 0.03$ and $\Delta t\, c^2 = 0.03$. The maximum divergence of the velocity field after 500 iterations was $|\nabla \bullet \mathbf{V}| < 5 \times 10^{-5}$ and the maximum change of the velocity components were less than 10^{-7}. The algorithm required 0.77 sec. of CPU time per full ADI step on SUN microsystems (Sparc 1^+).

The Zero Perturbation Method described above has a consistent accuracy for all solution variables. **Table II-I** reproduces the expected values of convergence rate α, i.e., for second-order approach $\alpha = 2$, based on three different grids. The value of κ represents the curvature of the initial interface.

The algorithm, as presented for small Reynolds limit, is easily upgraded to a more general class of flows. There are some advantages of this asymptotic approach. First all, its procedure of solving the field equations is completely the same as the Picard approach. Picard method is suitable for the solution of the free surface flow with an arbitrary surface tension. It involves an iterative process leading to the determination of the form of the interface and flow field simultaneously which is slightly different from present approach for solving the normal stress equation: in the Picard approach a form of the interface is assumed, then the governing equations are solved without enforcing the normal stress boundary condition at the interface which is subsequently used to deduce a new guess for the form of the interface. Secondly, the more general consideration of

TABLE II-I

Convergence Rates for Free Surface Flow with Curved Interface

Curved Interface $\kappa = 1.088$	**u**	**v**	**p**	**h**
G20-G30-G40	1.97	1.98	2.14	2.00
G20-G50-G80	1.99	1.99	2.04	2.00
Flat Interface $\kappa = 0.0$	**u**	**v**	**p**	**h**
G30-G50-G80	1.99	2.01	1.96	2.00
G40-G80-G100	1.99	2.01	2.07	2.00
Curved Interface $\kappa = -1.088$	**u**	**v**	**p**	**h**
G30-G40-G50	1.94	2.03	2.08	2.00
G20-G40-G80	2.00	1.99	2.11	2.00

Note: The first column gives the nubmer of grid points used in calculating the convergence rate. The number of grid points in x and y directions is the same.

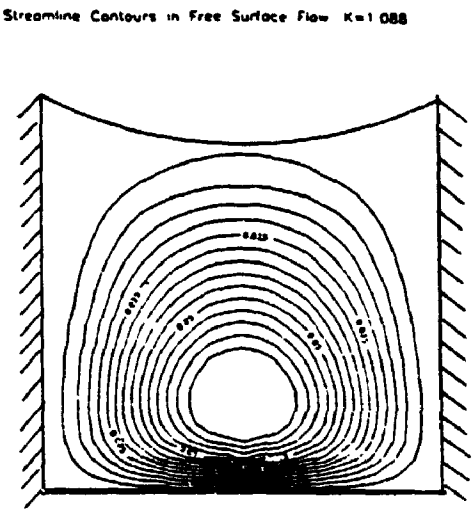
small or medium Reynolds number flows is not very difficult except the complication of the algebraic system of equations involved. In that case, the solutions will not be symmetric.

There is a limitation on the performance of the algorithm. The convergence was difficult to obtain for the large curvature, i.e., $\kappa > 1.088$ (where $-2 < \kappa < 2$), where the corner singularities appear in the two upper corners at the contact points. This may attribute to the fact that the shear stress is discontinuous at the contact point. The alternative assumption of the contact line with the solid wall is necessary for this problem since the local similarity analysis indicate the infinite pressure occurs at the contact point if the contact point is attached at the wall. We may also use the local analytical solutions to incorporate the numerical algorithm for the accurate and stable solution.

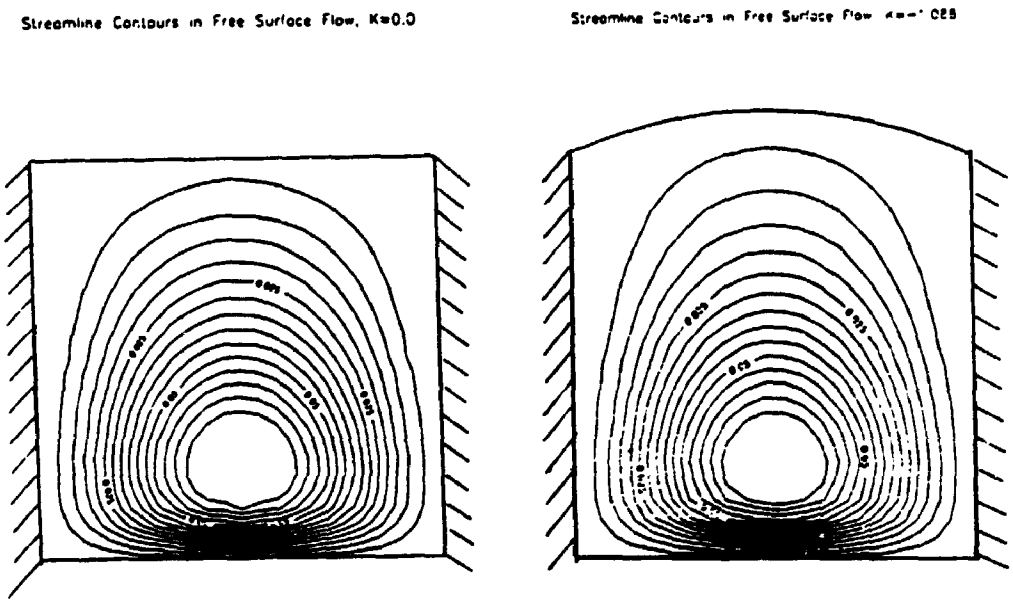
4.2 The Flow Patterns

The system of the field equations (II-45)-(II-47) and boundary conditions (II-48)-(II-52) essentially involves with a meniscus problem in which the interface has to be a surface of constant curvature. Consequently, this problem becomes a cavity flow problem with the upper surface rigid. Physically, we may interpret it as the case of strong capillary effects, i.e., large surface tension. The structures of such flow field will be given below.

Streamlines associated with the flow field in the curvatures of $\kappa = 1.088$, $\kappa = 0.0$ and $\kappa = -1.088$ are shown in **Figure 2.3**. The flow is driven by a velocity of $16[x(1-x)]^2$ at the bottom from left to right. When the fluid approaches the right wall, it turns towards the free surface and flows back along the surface line forming a vortex,



(a) Surface Bulge-in



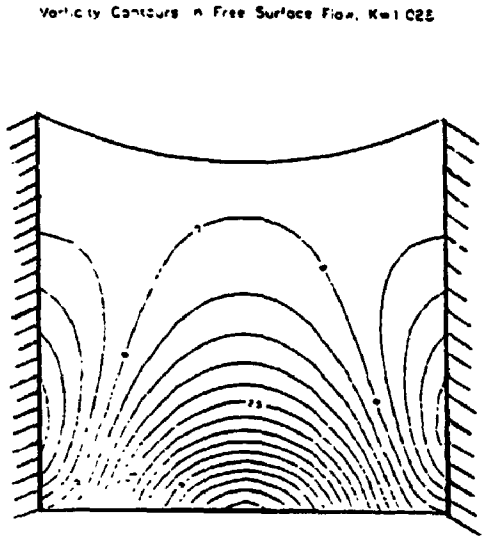
(b) Surface Flat

(c) Surface Bulge-out

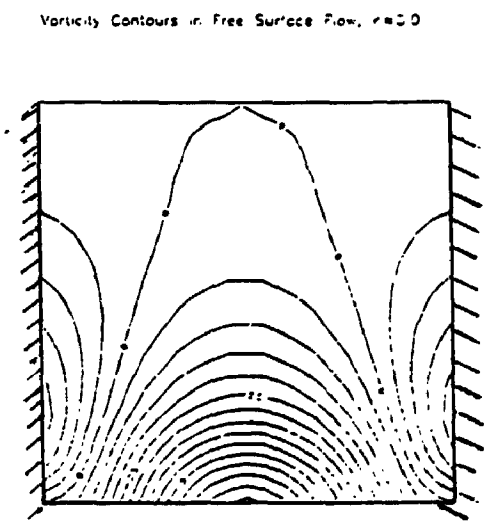
Figure 2.3 Streamline Flow Patterns for the Free Surface Flow with a Rigid Top. K's Represent the Constant Curvatures.

symmetric with respect to a vertical line passing through the centre. The centre of the vortex is located approximately one fifth of its height above the bottom line.

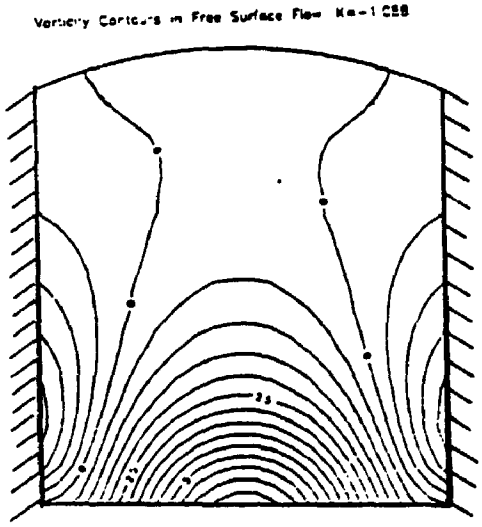
Figure 2.4 and **Figure 2.5** are the vorticity contours and surfaces respectively. The maximum vorticity occurs at the bottom of the cavity. The maximum values are $\omega_{max} = 7.4013$, $\omega_{max} = 7.2047$ and $\omega_{max} = 7.0377$ for the cases of $\kappa = 1.088$, $\kappa = 0.0$ and $\kappa = -1.088$ respectively, in which the cases represent the interface of bulge-in, flat and bulge-out. The vorticity distributions for the three cases are almost similar except different near the top. The different geometry causes the different vorticity distribution along the free surface line. The values of the vorticities on the free surface change from negative to positive for the three cases, i.e., at the centre of the surface $\omega_{in} = -0.2365$, $\omega_{flat} = -0.0244$ and $\omega_{out} = 0.0718$. It can be seen from the contour figure that the zero vorticity lines for bulge-out case are "open" along the surface line while others are "closed". The pressure contours and distribution surfaces are shown in **Figure 2.6** and **Figure 2.7**. The high variations of pressure occur at the bottom of the cavity. The maximum pressure is at the right bottom corner and minimum at the left. Their maximum values for three cases are $P_{in} = 9.726$, $P_{flat} = 9.715$ and $P_{out} = 9.727$ while the minimum values are the negative signs of the cases due to symmetry. Both pressure distributions are higher when the constant curvature is not equal to zero. The pressure distribution along the surface line are quite different with bulge-in and bulge-out. The former has smaller variations and pressures decrease from right to left. However, in the bulge-out case, its pressure changes are relatively higher and pressures increase from right to left.



(a) Surface Bulge-in

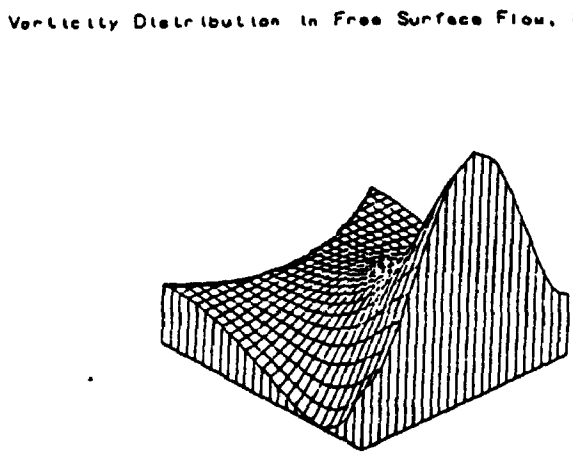


(b) Surface Flat

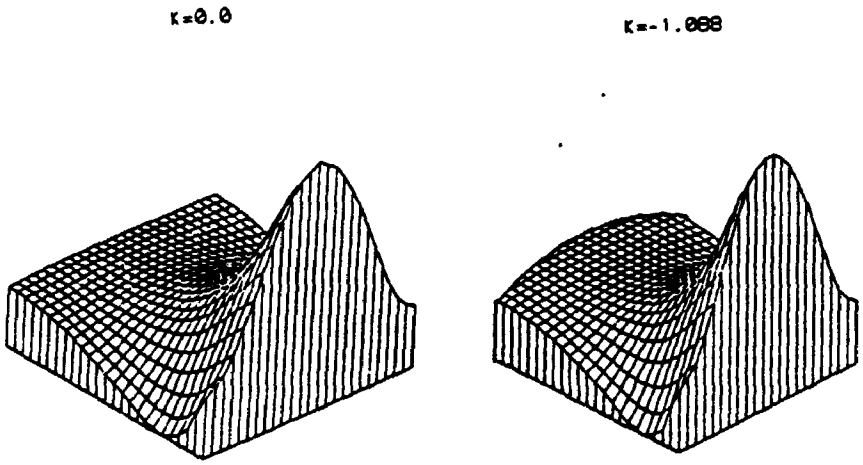


(c) Surface Bulge-out

Figure 2.4 Vorticity Contours in the Free Surface Flow with a Rigid Top.

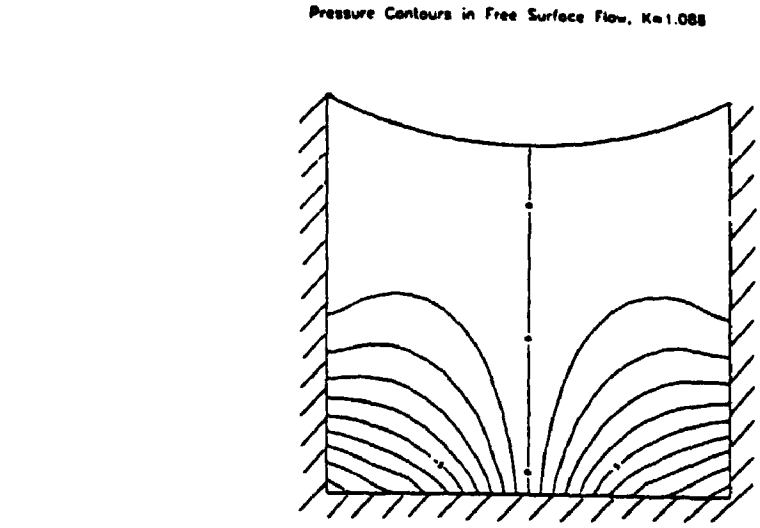


(a) Surface Bulge-in

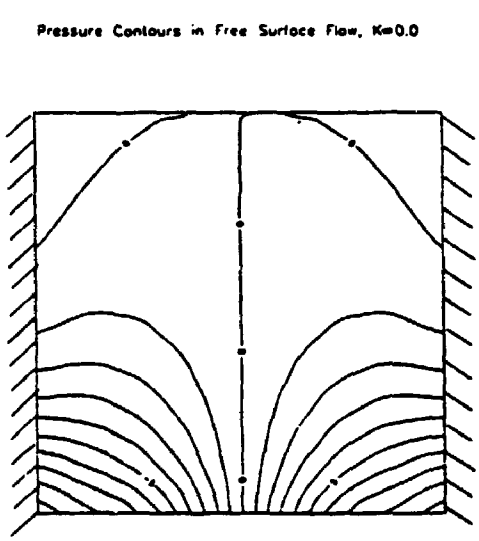


(b) Surface Flat *(c) Surface Bulge-out*

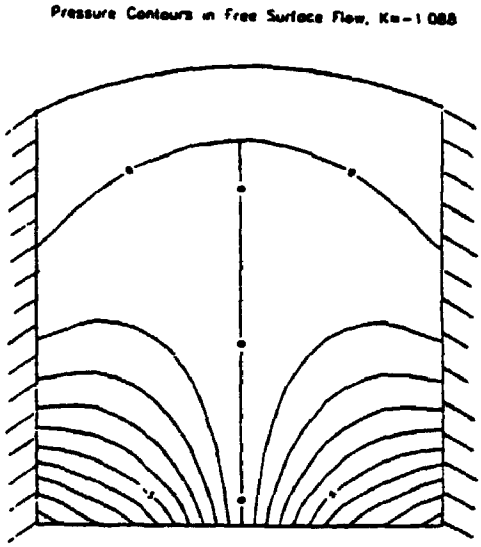
Figure 2.5 Vorticity Surfaces in the Free Surface Flow with a Rigid Top.



(a) Surface Bulge-in

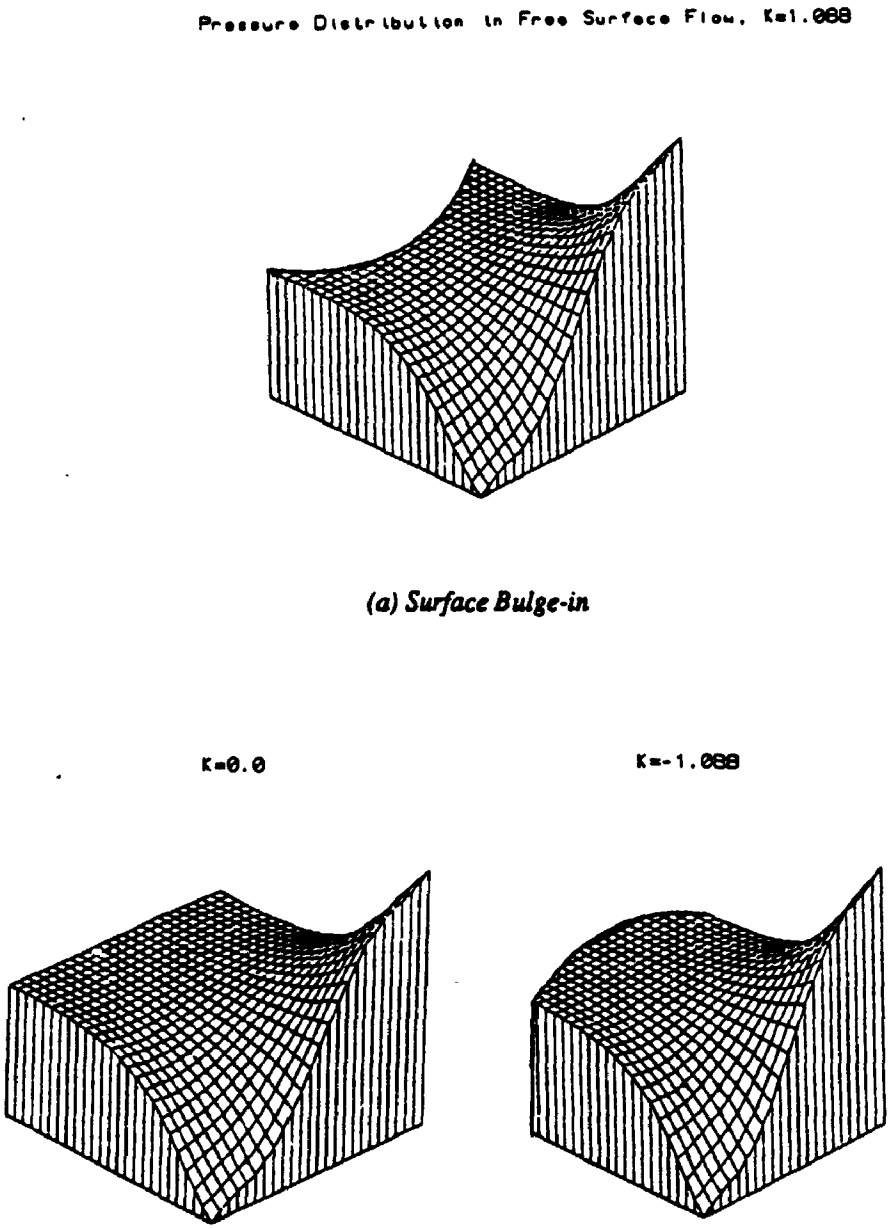


(b) Surface Flat



(c) Surface Bulge-out

Figure 2.6 Pressure Contours in the Free Surface Flow with a Rigid Top.



(a) Surface Bulge-in

(b) Surface Flat

(c) Surface Bulge-out

Figure 2.7 Pressure Surfaces in the Free Surface Flow with a Rigid Top.

4.3 The Deformation of Free Surface

Equation (II-58) describing surface deformation after transformation can be written in the form of

$$h_{1xx} + G(x)\, h_{1x} = H(x,p)\ , \tag{II-77}$$

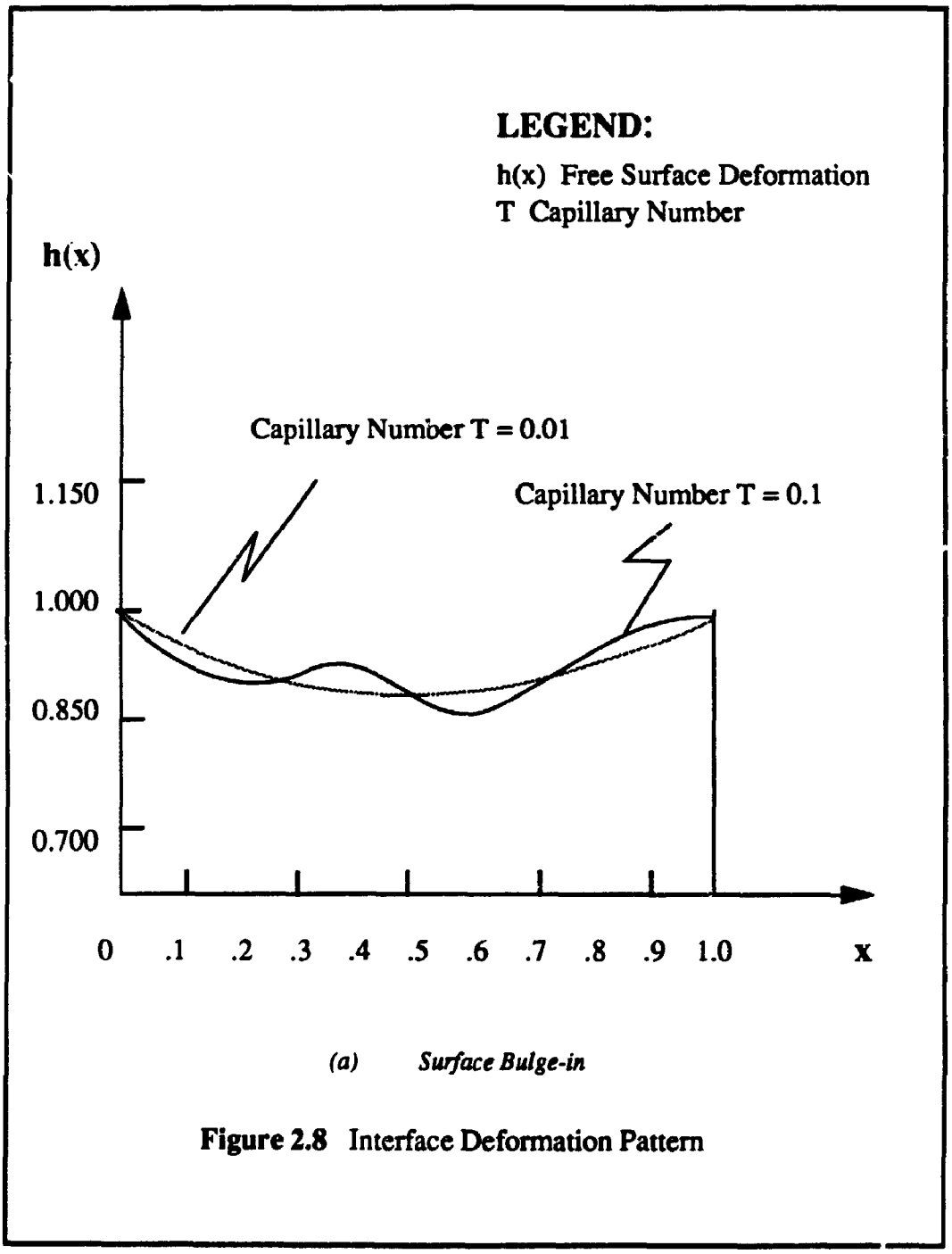
where

$$G(x) = -3h_{0x}h_{0xx}\left(1+h_{0x}^2\right)^{-1} \tag{II-78}$$

and

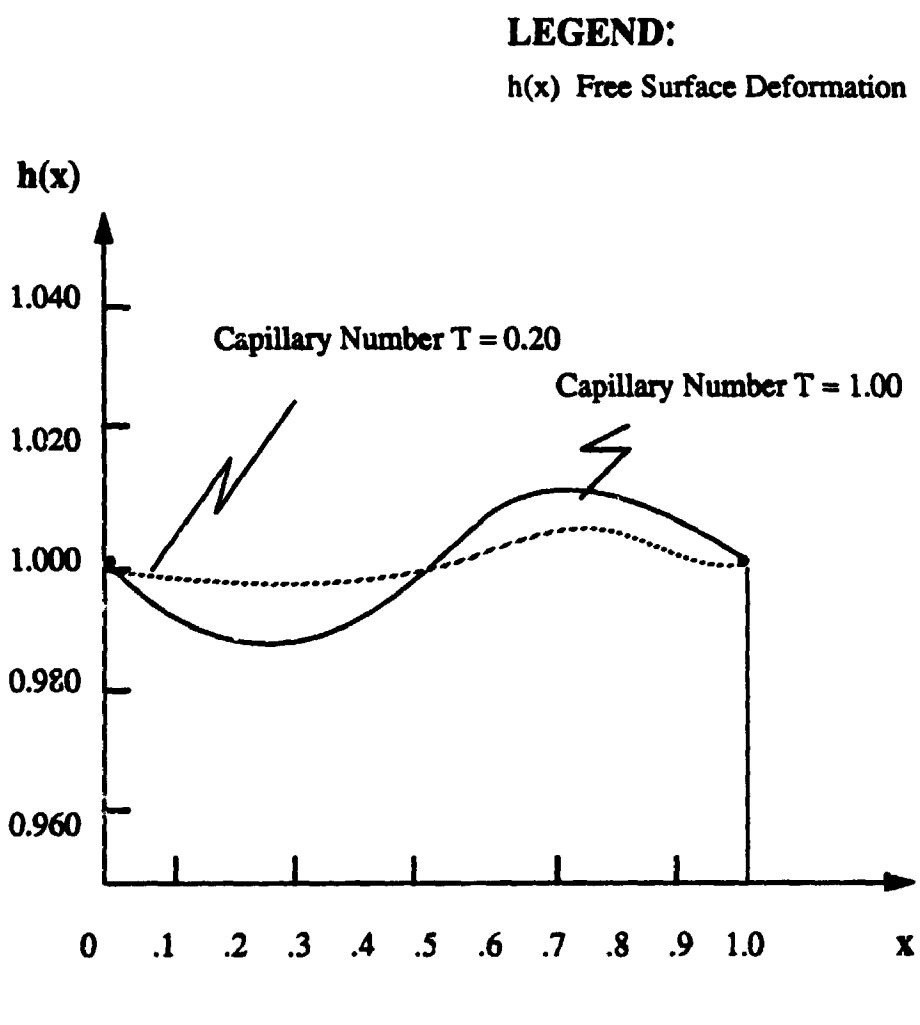
$$H(x,p) = -p+\bar{c}+\frac{2}{h\left(1+h_x^2\right)}\left\{\left(v_z-h_x u_z\right)+h_x\left(-hv_x+zh_x v_z+h_x\left(hu_x-zh_x u_z\right)\right)\right\} \tag{II-79}$$

In the above, p stands for the pressure field determined numerically from the field equations. There is an additive constant $\bar{c}$ included in pressure p that still needs to be determined. The appropriate boundary conditions are either the fixed contact line conditions (II-57) or the fixed angle conditions. In addition, the solution has to satisfy the volumetric constraint condition (II-56) which gives the extra equation for the constant $\bar{c}$. We note that (II-77) with its boundary conditions (II-56)-(II-57) are valid for arbitrary Reynolds numbers. Our conclusions regarding the qualitative form of the surface deformation are generally valid, however our quantitative results are limited only to the case of Stokes flow, as discussed throughout Part II. Our major interests are on the variations of surface deformation h_1 as function of the curvature κ and the different values of the capillary numbers ε (or T).



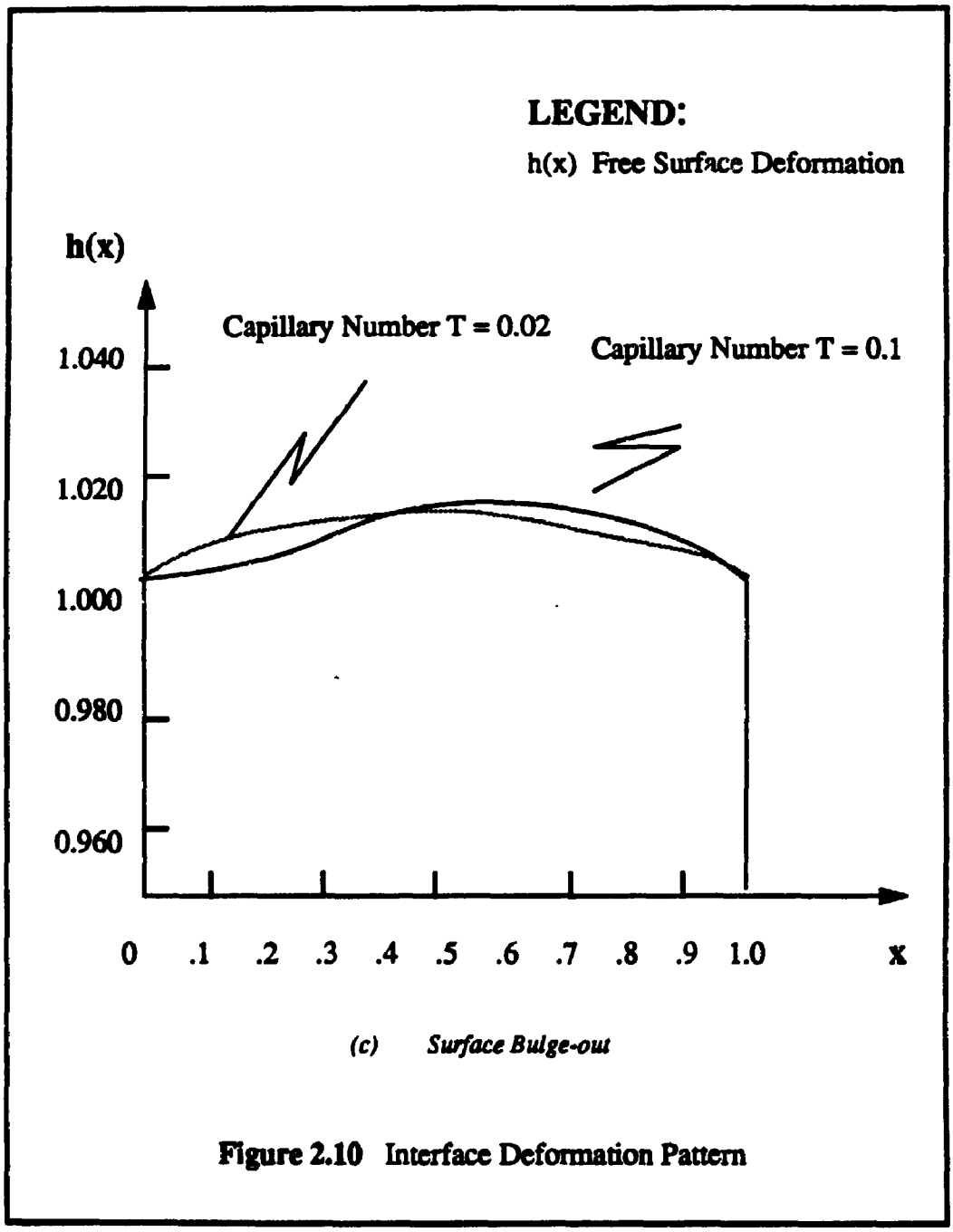
(a) Surface Bulge-in

Figure 2.8 Interface Deformation Pattern



(b) Surface Flat

Figure 2.9 Interface Deformation Pattern



(c) Surface Bulge-out

Figure 2.10 Interface Deformation Pattern

The numerical algorithm involves the numerical determination of the surface deformation h_1 incorporated with an integral equation (II-56) by discrete evaluation. We have selected the trapezoidal rule to evaluate the integral. Numerical approximation associated with such a procedure are second-order accuracy and is consistent with the error of the algorithm used to determine the required flow fields.

Figure 2.8, Figure 2.9 and **Figure 2.10** illustrate surface deformation of the cavity flow with the cases of bulge-in, flat and bulge-out for two different capillary numbers. The deformation is anti-symmetric with right side bulging out and left buldging in. Such deformation is a result of three competing effects. The first one is a dynamic effect associated with the motion of the liquid. The induced pressure gradient gives rise to a higher pressure at the bottom from left to right which causes the fluid to circulate. The second effect is a constant effect of curvature pressure caused by the curvature of the interface. This effect is relatively small in the present calculations. The third effect arises due to variations in surface tension that results in a capillary pressure gradient. The dynamic effect is strongest in the present case and leads to a bulge at the left top of the free surface.

The present results are only providing the moderate cases of the capillary number, curvature of the interface and driven velocities since the purpose of this thesis is to test the accuracy of the numerical algorithm. It will be interesting to carry some numerical calculations on the extreme cases in which it may be able to predict break-up of the interface. The asymptotic approach is only valid for small capillary numbers, therefore, direct solution is also necessary for solving this type of flow problems.

CHAPTER 5: SUMMARY

We have provided a detailed description of our accurate and reliable algorithm for simulation of moving capillary surfaces by using the Zero Perturbation Method in which the pressure problem has been completely eliminated. In chapter two we have described a model problem that is used to test our algorithm. In chapter three we have described a solution procedure valid in the limit of large surface tension. The algorithms used to solve field equations have been tested and its accuracies were verified through convergence rate analysis. The numerical results were found to be in good agreement with the corresponding theory discussed in Part I.

Throughout this part of the thesis, we have carried out an asymptotic approach of the free surface flow in a driven cavity in the absence of gravitational forces in the limit of small Capillary number. Such a situation arises when the surface tension is very large as compared to all other physical variables. The flow motion is considered to be steady in the present case. The small deformation theory is applied to decouple surface deformation from the field equations describing the motion of the liquid. The leading order of the approximation for the Young-Laplace equation (or normal stress condition) is obtained. The detailed analysis of the structure of the flow field as well as the surface deformation with the variation of the capillary number are given.

APPENDIX A: Programs for the Navier-Stokes Equations

```
**********************************************************************
*       MAIN PROGRAM FOR SOLVING THE DRIVEN CAVITY FLOW
*       USING STREAM FUNCTION-VORTICITY FORMULATION
*       MA'S UPWIND SCHEME IS USED FOR HIGH REYNOLDS FLOW
**********************************************************************

    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    PARAMETER (LN=21)
    DIMENSION B(LN),D(LN),E(LN),XSI0(LN,LN),
   1 PSI0(LN,LN),XSI(LN,LN),PSI(LN,LN),
   1 FPSI(LN),FXSI(LN),X(LN),Z(LN)
    DIMENSION B1(LN),D1(LN),E1(LN)
    DATA EER,RE/1E-6,100/
    OPEN(2,FILE='s.dat0')
    M=21
    N=21
    WRITE(*,*) 'M=? N=? KITE=? RE=? AF=? W=?'
    READ(*,*) M,N,KITE,RE,AF,W
    DX=1D0/(M-1D0)
    DZ=1D0/(N-1D0)
 63  FORMAT(3F12.7,1X,E12.6)
    WRITE(*,63)DX,DZ,RE,EER
    DO 900 I=1,M
900  X(I)=(I-1)*DX
```

```
      DO 901 J=1,N
  901 Z(J)=(J-1)*DZ
      DX1=1D0/DX
      DZ1=1D0/DZ
      DX2=1D0/DX/DX
      DZ2=1D0/DZ/DZ
c
      DO 903 I=1,M
      DO 904 J=1,N
      PSI(I,J)=0D0
      XSI(I,J)=0D0
  904 CONTINUE
  903 CONTINUE
c
C     DO 905 J=1,N
C     DO 905 I=1,M
C     READ(2,*)PSI(I,J),XSI(I,J)
C 905 CONTINUE
      DO 911 I=1,M
      D1(I)=0D0
      B1(I)=0D0
      E1(I)=0D0
      D(I)=0D0
      B(I)=0D0
  911 E(I)=0D0

c
      ITE=1
  90  CONTINUE

c
```

```
      MOD2=MOD(ITE,KITE)
      IF (MOD2.EQ.0) THEN
      DO 737 I=1,M
      DO 738 J=1,N
      PSI0(I,J)=PSI(I,J)
      XSI0(I,J)=XSI(I,J)
  738 CONTINUE
  737 CONTINUE
      END IF
C
        MB=2
        NB=2
        NE=N-1
        ME=M-1

C       BOUNDARY CONDITIONS: J=2 OR N-1
        DO 1005 I=2,M-1
        XSI(I,2)=-(-2D0*PSI(I,2)+PSI(I,1)+PSI(I,3))*DZ2
     1    -(-2D0*PSI(I,2)+PSI(I-1,2)+PSI(I+1,2))*DX2
        PSI(I,2)=PSI(I,3)/4D0
 1005 CONTINUE

      DO 1777 I=3,M-2
      D(I)=DX2
      B(I)=-2D0*DX2-2D0*DZ2
      E(I)=DX2
 1777 CONTINUE

      DO 1001 J=3,N-2
       JM1=J-1
       JP1=J+1
```

```
      DO 2001 I=3,M-2
      IP1=I+1
      IM1=I-1
      U=(PSI(I,JP1)-PSI(I,JM1))*DZ1/2D0
      V=-(PSI(IP1,J)-PSI(IM1,J))*DX1/2D0
      IF (U.GE.0) THEN
      D1(I)=DZ2
      B1(I)=-2D0*DZ2-2D0*DX2-RE/W+(1-AF)*RE*DX1*U/2D0
      E1(I)=DZ2-(1-AF)*RE*DX1*U/2D0
      FXSI(I)=-(XSI(I,JP1)+XSI(I,JM1))*DZ2-RE*XSI(I,J)/W
     1  +(1+AF)*RE*DX1*U*(XSI(I,J)-XSI(IM1,J))/2D0
      ELSE
      E1(I)=DZ2
      B1(I)=-2D0*DZ2-2D0*DX2-RE/W-(1+AF)*RE*DX1*U/2D0
      D1(I)=DZ2+(1+AF)*RE*DX1*U/2D0
      FXSI(I)=-(XSI(I,JP1)+XSI(I,JM1))*DZ2-RE*XSI(I,J)/W
     1  +(1-AF)*RE*DX1*U*(XSI(IP1,J)-XSI(I,J))/2D0
      ENDIF
      FXSI(I)=FXSI(I)
     1  +(1-AF)*RE*DZ1*V*(XSI(I,JP1)-XSI(I,J))/2D0
     1  +(1+AF)*RE*DZ1*V*(XSI(I,J)-XSI(I,JM1))/2D0

      FPSI(I)=-(PSI(I,JP1)+PSI(I,JM1))*DZ2
     1        -XSI(I,J)
2001  CONTINUE
      D(2)=0D0
      B(2)=-2D0*DX1
      E(2)=DX1/2D0
      D(M-1)=-DX1/2D0
      B(M-1)=2D0*DX1
      E(M-1)=0D0
```

```
      FPSI(2)=0D0
      FPSI(M-1)=0D0
      CALL THOMSX(D,B,E,FPSI,J,M,N,MB,ME,PSI)
      D1(2)=0D0
      B1(2)=1D0
      E1(2)=0D0
      D1(M-1)=0D0
      B1(M-1)=1D0
      E1(M-1)=0D0
      FXSI(2)=-(-2D0*PSI(2,J)+PSI(1,J)+PSI(3,J))*DZ2
     1    -(-2D0*PSI(2,J)+PSI(2,J-1)+PSI(2,J+1))*DX2
      FXSI(M-1)=-(-2D0*PSI(M-1,J)+PSI(M,J)+PSI(M-2,J))*DZ2
     1    -(-2D0*PSI(M-1,J)+PSI(M-1,J-1)+PSI(M-1,J+1))*DX2
      CALL THOMSX(D1,B1,E1,FXSI,J,M,N,MB,ME,XSI)
 1001 CONTINUE
C
      DO 1006 I=2,M-1
      XSI(I,N-1)=-(-2D0*PSI(I,N-1)+PSI(I,N-2)+PSI(I,N))*DZ2
     1    -(-2D0*PSI(I,N-1)+PSI(I-1,N-1)+PSI(I+1,N-1))*DX2
      PSI(I,N-1)=-DX/2D0+PSI(I,N-2)/4D0
 1006 CONTINUE

C     Z -SWEEP:

      DO 1778 J=3,N-2
      D(J)=DX2
      B(J)=-2D0*DX2-2D0*DZ2
      E(J)=DX2
 1778 CONTINUE
      DO 1007 J=2,N-1
      XSI(2,J)=-(-2D0*PSI(2,J)+PSI(1,J)+PSI(3,J))*DZ2
```

```
     1    -(-2D0*PSI(2,J)+PSI(2,J-1)+PSI(2,J+1))*DX2
          PSI(2,J)=PSI(3,J)/4D0
     1007 CONTINUE

C
          DO 3010 I=3,M-2
          IP1=I+1
          IM1=I-1
          DO 4100 J=3,N-2
          JP1=J+1
          JM1=J-1
          U=(PSI(I,J+1)-PSI(I,J-1))*DZ1/2D0
          V=-(PSI(I+1,J)-PSI(I-1,J))*DX1/2D0
          IF (V.GE.0) THEN
          D1(J)=DX2
          B1(J)=-2D0*DZ2-2D0*DX2-RE/W+(1-AF)*RE*DZ1*V/2D0
          E1(J)=DX2-(1-AF)*RE*DZ1*V/2D0
          FXSI(J)=-(XSI(IP1,J)+XSI(IM1,J))*DX2-RE*XSI(I,J)/W
     1    +(1+AF)*RE*DZ1*V*(XSI(I,J)-XSI(I,JM1))/2D0
          ELSE
          E1(J)=DX2
          B1(J)=-2D0*DZ2-2D0*DX2-RE/W-(1+AF)*RE*DZ1*V/2D0
          D1(J)=DX2+(1+AF)*RE*DZ1*V/2D0
          FXSI(J)=-(XSI(IP1,J)+XSI(IM1,J))*DX2-RE*XSI(I,J)/W
     1    +(1-AF)*RE*DZ1*V*(XSI(I,JP1)-XSI(I,J))/2D0
          ENDIF
          FXSI(J)=FXSI(J)
     1    +(1-AF)*RE*DX1*U*(XSI(IP1,J)-XSI(I,J))/2D0
     1    +(1+AF)*RE*DX1*U*(XSI(I,J)-XSI(IM1,J))/2D0
          FPSI(J)=-(PSI(IP1,J)+PSI(IM1,J))*DZ2
     1         -XSI(I,J)
```

```
4100  CONTINUE
      D(2)=0D0
      B(2)=-2D0*DX1
      E(2)=DX1/2D0
      D(N-1)=-DX1/2D0
      B(N-1)=2D0*DX1
      E(N-1)=0D0
      FPSI(2)=0D0
      FPSI(N-1)=-1D0
      CALL THOMSY(D,B,E,FPSI,I,M,N,NB,NE,PSI)
      D1(2)=0D0
      B1(2)=1D0
      E1(2)=0D0
      D1(N-1)=0D0
      B1(N-1)=1D0
      E1(N-1)=0D0
       FXSI(2)=-(-2D0*PSI(I,2)+PSI(I,1)+PSI(I,3))*DZ2
     1   -(-2D0*PSI(I,2)+PSI(I-1,2)+PSI(I+1,2))*DX2
       FXSI(N-1)=-(-2D0*PSI(I,N-1)+PSI(I,N-2)+PSI(I,N))*DZ2
     1   -(-2D0*PSI(I,N-1)+PSI(I-1,N-1)+PSI(I+1,N-1))*DX2
      CALL THOMSY(D1,B1,E1,FXSI,I,M,N,NB,NE,XSI)
3010  CONTINUE

      DO 1008 J=2,N-1
      XSI(M-1,J)=-(-2D0*PSI(M-1,J)+PSI(M,J)+PSI(M-2,J))*DZ2
     1   -(-2D0*PSI(M-1,J)+PSI(M-1,J-1)+PSI(M-1,J+1))*DX2
      PSI(M-1,J)=PSI(M-2,J)/4D0
1008  CONTINUE
C
C
      IF (ITE.EQ.1) SE=1D0
```

```
      IF (MOD2.EQ.0) THEN
      SE=0D0
      DO 727 I=2,M-1
      DO 728 J=2,N-1
      SE1=DABS(PSI(I,J)-PSI0(I,J))
      IF (SE1.GT.SE) SE=SE1
  728 CONTINUE
  727 CONTINUE
      DO 127 I=2,M-1
      DO 128 J=2,N-1
      SE1=DABS(XSI(I,J)-XSI0(I,J))
      IF (SE1.GT.SE) SE=SE1
  128 CONTINUE
  127 CONTINUE
      END IF
      IF (MOD2.EQ.0) THEN
      WRITE(6,81)SE,ITE
   81 FORMAT(10X,'SE=',E16.10,20X,'ITE=',I9)
      END IF
      IF (SE.LT.EER) GOTO 95
C
      MOD3=MOD(ITE,10*KITE)
      IF(MOD3.EQ.0) THEN
      DO 7223 J=1,N
      WRITE(2,'(10X)')
      DO 7222 I=1,M
      WRITE(2,4141)PSI(I,J),XSI(I,J)
 7222 CONTINUE
 7223 CONTINUE
      REWIND 2
      END IF
```

```
      ITE=ITE+1
c
      GOTO 90
c
c
 95   CONTINUE
      DO 420 I=2,M-1
      XSI(I,1)=(2D0*XSI(I,2)-XSI(I,3))
      XSI(I,M)=(2D0*XSI(I,M-1)-XSI(I,M-2))
 420  CONTINUE
      DO 430 J=1,N
      XSI(1,J)=(2D0*XSI(2,J)-XSI(3,J))
      XSI(N,J)=(2D0*XSI(N-1,J)-XSI(N-2,J))
 430  CONTINUE

      DO 9223 J=1,N
      WRITE(2,'(10X)')
      DO 9222 I=1,M
      WRITE(2,4141) PSI(I,J),XSI(I,J)
9222  CONTINUE
9223  CONTINUE
4141  FORMAT(10X,D20.14,3X,D20.14)

      STOP
      END

      SUBROUTINE THOMSX(D,B,E,FUV,J,M,N,MB,ME,UV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (LN=21)
      DIMENSION D(LN),B(LN),E(LN),FUV(LN),UV(LN,LN),
     1  R(LN),S(LN)
```

```
      FMB=FUV(MB)
      FME=FUV(ME)
      FUV(MB)=FMB-D(MB)*UV(MB-1,J)
      FUV(ME)=FME-E(ME)*UV(ME+1,J)
      R(MB)=FUV(MB)/B(MB)
      S(MB)=-E(MB)/B(MB)
      DO 2543 K=MB+1,ME
      KM1=K-1
      RS=B(K)+D(K)*S(KM1)
      R(K)=(FUV(K)-D(K)*R(KM1))/RS
2543    S(K)=-E(K)/RS
      UV(ME,J)=R(ME)
      DO 2544 I1=MB,ME-1
      K=ME-1+MB-I1
2544    UV(K,J)=UV(K+1,J)*S(K)+R(K)
      RETURN
      END

      SUBROUTINE THOMSY(D,B,E,FUV,I,M,N,NB,NE,UV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (LN=21)
      DIMENSION D(LN),B(LN),E(LN),FUV(LN),UV(LN,LN),
     1  R(LN),S(LN)
      FNB=FUV(NB)
      FNE=FUV(NE)
      FUV(NB)=FNB-D(NB)*UV(I,NB-1)
      FUV(NE)=FNE-E(NE)*UV(I,NE+1)
      R(NB)=FUV(NB)/B(NB)
      S(NB)=-E(NB)/B(NB)
      DO 2543 K=NB+1,NE
```

```
      KM1=K-1
      RS=B(K)+D(K)*S(KM1)
      R(K)=(FUV(K)-D(K)*R(KM1))/RS
2543    S(K)=-E(K)/RS
      UV(I,NE)=R(NE)
      DO 2544 I1=NB,NE-1
      K=NE-1+NB-I1
2544    UV(I,K)=UV(I,K+1)*S(K)+R(K)
      RETURN
      END
```

APPENDIX B: Programs for Free Surface Flows

```
************************************************************************
*       MAIN PROGRAM FOR SOLVING THE FREE SURFACE FLOW
*      LARGE SURFACE TENSION AND SMALL REYNOLDS LIMIT CASE
*       NUMERICAL METHOD :  ASYMTOTIC APPROACH
*       T  ----  CAPILLARY NUMBER
*       H  ----  INITIAL POSITION OF INTERFACE
*       H1  ----  CORRECTION OF SURFACE DEFORMATION
*       Vm  ----  GIVEN VOLUME TO DETERMINE INITIAL INTERFACE H
*       PS0 ----  CURVATURE OF THE INITIAL INTERFACE
************************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21,M1=21)
      DIMENSION B(N1),D(N1),E(N1),H1(M1),
     1 FAN(M1),FBN(M1),FCN(M1),FU(N1),FV(N1),
     1 US(M1,N1),VS(M1,N1),PS(M1,N1)
      COMMON /U/U(M1,N1)/V/V(M1,N1)
     1 /H/H(M1),HH(M1)/HX/HX(M1),HXX(M1),HXHX(M1),
     1  HXXH(M1),HXH(M1)/P/P(M1,N1)
     1 /XZ/X(M1),Z(M1)/DXZ/DX,DZ/WW/W
     1 /DPB/PX1J(M1),PXMJ(M1),PZ1J(M1),PZMJ(M1),
     1 PXI1(M1),PXIN(M1),PZI1(M1),PZIN(M1)
      DATA HL,HR,W,CP,CN,KX,M,N/1D0,1D0,1D0,1D0,1D0,0,21,21/
      OPEN(1,FILE='s.dat0')
```

```
      WRITE(6,1)
    1  FORMAT(15X,' T=? DT=?  EER=?  VM=?  KITE=?'//)
   13  FORMAT(15X,' DX=?  T=?  DT=?  ERR=? Vm=?  KITE=?'//)
      WRITE(2,13)
   63  FORMAT(2F14.7,1X,E12.5,5X,F10.5,2X,E10.4/)
      READ(*,*)T,DT,EER,VM,KITE
      DX=1D0/(DBLE(M)-1D0)
      DZ=1D0/(DBLE(N)-1D0)
      WRITE(2,63)DX,CP,DT,EER,W,VM
      WRITE(6,63)DX,CP,DT,EER,W,VM
      WRITE(2,'(25X,3HKK=,I9//)')KITE
      KMM=INT(1D-1/DX)
      DO 900 I=1,M
  900 X(I)=(I-1)*DX
      DO 901 J=1,N
  901 Z(J)=(J-1)*DZ
      LI=(M-1)/2+1+KX
      LJ=(N-1)/2+1+KX
      PI=4D0*DATAN(1D0)
      DX1=1D0/DX
      DZ1=1D0/DZ
      DX2=1D0/DX/DX
      DZ2=1D0/DZ/DZ
      H(1)=HL
      H(M)=HR
      DO 119 I=2,M-1
  119 H(I)=1D0
      PS0=0D0

**********************************************************************
*     SUBROUTINE H0PS SOLVES H(I) AND PS0 WITH GIVEN VALUES
```

```
*    OF HL,HR (HIGHT OF THE CAVITY) AND VM (VOLUME OF LIQUID)
**********************************************************************

     CALL H0PS(H,PS0,VM,HL,HR,DX,M)

**********************************************************************
*    READ INITIAL U,V,P FROM FILE OR CALCULATED BY ITSELF
**********************************************************************
     DO 903 J=1,N
     DO 904 I=1,M
     READ(1,*)U(I,J),V(I,J),P(I,J)
 904  CONTINUE
 903  CONTINUE
c
c    DO 903 I=1,M
c    UIJ0=16D0*X(I)*X(I)*(1D0-X(I))*(1D0-X(I))
c    DO 904 J=1,N
c    U(I,J)=UIJ0
c    V(I,J)=0D0
c    P(I,J)=1D0
c904  CONTINUE
c903  CONTINUE

**********************************************************************
*    CALCULATE THE INITIAL VALUES, I.E., PRESSURE DERIVATIVES
*         THE DERIVATIVES OF H(I) ARE FIXED THROUGHOUT
*         THE ITERATION.
**********************************************************************
     ITE=0
c
c
```

```
      CALL PRESSURE(DT,M,N,CP,DX1,DZ1,ITE)

      ITE=1
c
      HX(1)=(4D0*H(2)-3D0*H(1)-H(3))*DX1/2D0
      HX(M)=(3D0*H(M)+H(M-2)-4D0*H(M-1))*DX1/2D0
      DO 347 I=2,M-1
      HX(I)=(H(I+1)-H(I-1))*DX1/2D0
      HXX(I)=(H(I+1)-2D0*H(I)+H(I-1))*DX2
  347 CONTINUE
      HXX(1)=(2D0*H(1)-5D0*H(2)+4D0*H(3)-H(4))*DX2
      HXX(M)=(2D0*H(M)-5D0*H(M-1)+4D0*H(M-2)-H(M-3))*DX2
      DO 348 I=1,M
      HH(I)=H(I)*H(I)
      HXH(I)=H(I)*HX(I)
      HXHX(I)=HX(I)*HX(I)
      HXXH(I)=HXX(I)*H(I)
      FAN(I)=(1D0-HXHX(I))*HXXH(I)
      FBN(I)=-(1D0+HXHX(I))*HXH(I)
      FCN(I)=(1D0+HXHX(I))*(1D0+HXHX(I))
  348 CONTINUE
c
**********************************************************************
*     LOOP STARTS HERE
**********************************************************************

   90 CONTINUE
c
      DO 737 I=1,M
      DO 738 J=1,N
      US(I,J)=U(I,J)
```

```
      PS(I,J)=P(I,J)
      VS(I,J)=V(I,J)
  738 CONTINUE
  737 CONTINUE
*****************************************************
*     X-SWEEP:
*****************************************************
       MM1=M-1
      DO 1001 J=2,N-1
       ZJ=Z(J)*Z(J)
       JM1=J-1
       JP1=J+1
      DO 2001 I=2,MM1
       IP1=I+1
       IM1=I-1
       UXZIJ=(U(IP1,JP1)+U(IM1,JM1)-U(IP1,JM1)-U(IM1,JP1))/DZ/DX/4D0
       VXZIJ=(V(IP1,JP1)+V(IM1,JM1)-V(IP1,JM1)-V(IM1,JP1))/DZ/DX/4D0
       PXIJ=(P(IP1,J)-P(IM1,J))*DX1/2D0
       PZIJ=(P(I,JP1)-P(I,JM1))*DZ1/2D0
       E1=HH(I)*DX2
       E2=ZJ*HXHX(I)*DZ2+DZ2
       E3=(2D0*Z(J)*HXHX(I)-Z(J)*HXXH(I))*DZ1
       E4=HH(I)*U(I,J)/DT-PXIJ*HH(I)+
     1   Z(J)*HXH(I)*PZIJ-2D0*Z(J)*HXH(I)*UXZIJ
       E5=HH(I)*V(I,J)/DT-PZIJ*H(I)-
     1   2D0*Z(J)*HXH(I)*VXZIJ
       D(I)=-E1
       B(I)=HH(I)/DT+2D0*E1+2D0*E2
       E(I)=-E1
       AU=(-E2+E3/2D0)*U(I,JM1)
       CU=(-E2-E3/2D0)*U(I,JP1)
```

```
      AV=(-E2+E3/2D0)*V(I,JM1)
      CV=(-E2-E3/2D0)*V(I,JP1)
      FU(I)=E4-AU-CU
      FV(I)=E5-AV-CV
 2001 CONTINUE
      CALL THOMASX(D,B,E,FU,J,M,N,U)
      CALL THOMASX(D,B,E,FV,J,M,N,V)
 1001 CONTINUE
*************************************************
*     BOUNDARY CONDITIONS
*************************************************
c
c       J=N
c
      DO 6122 I=2,M-1
      E1=(HXHX(I)+1D0)*DZ1/H(I)/2D0
      VX=(V(I+1,N)-V(I-1,N))*DX1/2D0
      VZ=(3D0*V(I,N)+V(I,N-2)-4D0*V(I,N-1))*DZ1/2D0
      D(I)=-HX(I)*DX1
      B(I)=-3D0*E1
      E(I)=HX(I)*DX1
      FU(I)=E1*(U(I,N-2)-4D0*U(I,N-1))+2D0*HX(I)*VZ/H(I)
     1      +(1D0-HX(I)*HX(I))*(VX-HX(I)*VZ/H(I))
c     U(I)=(D(I)*U(I-1,N)+E(I)*U(I+1,N)-FU(I))/B(I)
 6122 CONTINUE
      CALL THOMASX(D,B,E,FU,N,M,N,U)
      DO 2015 I=1,M
 2015 V(I,N)=HX(I)*U(I,N)
c
*************************************************
*     Z-SWEEP:
```

```
*************************************************
      DO 3010 I=2,MM1
       IP1=I+1
       IM1=I-1
      DO 2100 J=2,N-1
       JP1=J+1
       JM1=J-1
       UXZIJ=(U(IP1,JP1)+U(IM1,JM1)-U(IP1,JM1)-U(IM1,JP1))/DZ/DX/4D0
       VXZIJ=(V(IP1,JP1)+V(IM1,JM1)-V(IP1,JM1)-V(IM1,JP1))/DZ/DX/4D0
       PXIJ=(P(IP1,J)-P(IM1,J))*DX1/2D0
       PZIJ=(P(I,JP1)-P(I,JM1))*DZ1/2D0
       E1=HH(I)*DX2
       E2=Z(J)*Z(J)*HXHX(I)*DZ2+DZ2
       E3=(2D0*Z(J)*HXHX(I)-Z(J)*HXXH(I))*DZ1
       E4=HH(I)*U(I,J)/DT-PXIJ*HH(I)+
     1   Z(J)*HXH(I)*PZIJ-2D0*Z(J)*HXH(I)*UXZIJ
       E5=HH(I)*V(I,J)/DT-PZIJ*H(I)-
     1   2D0*Z(J)*HXH(I)*VXZIJ
       D(J)=-E2+E3/2D0
       B(J)=HH(I)/DT+2D0*E1+2D0*E2
       E(J)=-E2-E3/2D0
       DU=-E1*U(IM1,J)
       EU=-E1*U(IP1,J)
       DV=-E1*V(IM1,J)
       EV=-E1*V(IP1,J)
       FU(J)=E4-DU-EU
       FV(J)=E5-DV-EV
 2100  CONTINUE
*************************************************
*     BOUNDARY CONDITIONS
*************************************************
```

```
      U(I,1)=16*X(I)*X(I)*(1D0-X(I))*(1D0-X(I))
      FU(2)=FU(2)-D(2)*U(I,1)
      E1=(HX(I)*HX(I)+1D0)*DZ1/H(I)/2D0
      VX=(V(I+1,N)-V(I-1,N))*DX1/2D0
      VZ=(3D0*V(I,N)+V(I,N-2)-4D0*V(I,N-1))*DZ1/2D0
      DN=-E1
      BN=4D0*E1
      EN=-3D0*E1
      FUN=-HX(I)*(U(I+1,N)-U(I-1,N))*DX1+2D0*HX(I)*VZ/H(I)
     1    +(1D0-HX(I)*HX(I))*(VX-HX(I)*VZ/H(I))
      DV=D(N-1)
      BV=B(N-1)
      D(N-1)=EN*D(N-1)-DN*E(N-1)
      B(N-1)=EN*B(N-1)-BN*E(N-1)
      FU(N-1)=EN*FU(N-1)-FUN*E(N-1)
      CALL THOMASY(D,B,E,FU,I,M,N,U)
      U(I,N)=(FUN-DN*U(I,N-2)-BN*U(I,N-1))/EN
      V(I,N)=HX(I)*U(I,N)
      V(I,1)=0D0
      FV(N-1)=FV(N-1)-E(N-1)*V(I,N)
      D(N-1)=DV
      B(N-1)=BV
      CALL THOMASY(D,B,E,FV,I,M,N,V)
 3010 CONTINUE
C
C
C
C
*****************************************************************
*     UPDATING THE NEW PRESSURE USING ARTIFICIAL
```

```
*     COMPRESSIBILITY METHOD WITH MODIFIED
*     BOUNDARY CONDITION FOR PRESSURE
**********************************************************************
      CALL PRESSURE(DT,M,N,CP,DX1,DZ1,ITE)
**********************************************
*     CHECKING CONVERGENCE
**********************************************

      SE=0D0
      DO 727 I=2,M-1
      DO 728 J=2,N
      SE1=DABS(U(I,J)-US(I,J))
      IF (SE1.GT.SE) SE=SE1
728   CONTINUE
727   CONTINUE
      DO 127 I=2,M-1
      DO 128 J=2,N
      SE1=DABS(V(I,J)-VS(I,J))
      IF (SE1.GT.SE) SE=SE1
128   CONTINUE
127   CONTINUE
      DO 227 I=1,M
      DO 228 J=1,N
      SE1=DABS(P(I,J)-PS(I,J))
      IF (SE1.GT.SE) SE=SE1
228   CONTINUE
227   CONTINUE
      MOD3=MOD(ITE,KITE)
      IF (MOD3.EQ.0) THEN
      WRITE(6,81)SE,ITE
81    FORMAT(10X,'SE=',E16.10,20X,'ITE=',I9)
```

```
      END IF

      IF (SE.LT.EER) GOTO 95

      MOD2=MOD(ITE,10*KITE)
      IF(MOD2.EQ.0) THEN
      DO 7223 J=1,N
      WRITE(1,'(10X)')
      DO 7222 I=1,M
      WRITE(1,2111)U(I,J),V(I,J),P(I,J)
 7222 CONTINUE
 7223 CONTINUE
      REWIND 1
      END IF

      ITE=ITE+1

      GOTO 90

*****************************************************************
*     CALCULATING H1 CORRECTION OF SURFACE DEFORMATION
*     USING SUBROUTINE HORDER2 WHICH IS A SECOND
*     ORDER O.D.E.
*****************************************************************

  95  LM=(M-1)/2+1
      PIN=P(LM,N)
      DO 422 I=1,M
      PPIN=P(I,N)
      P(I,N)=PPIN-PIN
 422  CONTINUE
```

```
      CALL HORDER2(H1,M,N,CN)
      DO 500 I=1,M
      PPIN=P(I,N)
      P(I,N)=PPIN+PIN
  500 CONTINUE

*****************************************************
*     WRITING RESULTS
*****************************************************

      REWIND 1
      DO 5223 J=1,N
      WRITE(1,'(10X)')
      DO 5222 I=1,M
      WRITE(1,2111)U(I,J),V(I,J),P(I,J)
 5222 CONTINUE
 5223 CONTINUE
      DO 301 I=1,M
      WRITE(1,'(10X,F15.7)')H(I)
  301 CONTINUE
      WRITE(1,'(//10X,7HRESULT=//)')
      WRITE(1,'(10X,3HDX=,F15.7)')DX
      WRITE(1,'(/5X,4F15.7/)')T,5D0*T,10D0*T,50D0*T
      DO 4223 I=1,M
      WRITE(1,2101)(H(I)+H1(I)*T),(H(I)+H1(I)*T*5D0),
     1          (H(I)+H1(I)*T*10D0),(H(I)+H1(I)*T*50D0)
 4223 CONTINUE
 2101 FORMAT(3X,4F15.9)
 2111 FORMAT(10X,D20.14,3X,D20.14,3X,D20.14)
      STOP
      END
```

```
**********************************************
*    SUBROUTINES START HERE
**********************************************

      SUBROUTINE COEFF(W,A,B,K0)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /DXZ/DX,DZ
      IF (K0.EQ.1) THEN
      A=(2D0*DX-2D0*W)/(DX*DX)
      B=-(DX-2D0*W)/(2*DX*DX)
      END IF
      IF (K0.EQ.-1) THEN
      A=-(2D0*DX-2D0*W)/(DX*DX)
      B=(DX-2D0*W)/(2*DX*DX)
      END IF
      RETURN
      END

      SUBROUTINE PRESSURE(DT,M,N,CP,DX1,DZ1,ITE)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21,M1=21)
      COMMON /U/U(M1,N1)/V/V(M1,N1)/H/H(M1),HH(M1)
     1 /HX/HX(M1),HXX(M1),HXHX(M1),HXXH(M1),HXH(M1)
     1 /P/P(M1,N1)/XZ/X(M1),Z(N1)/WW/W/DXZ/DX,DZ
     1 /DPB/PX1J(N1),PXMJ(N1),PZ1J(N1),PZMJ(N1),
     1 PXI1(M1),PXIN(M1),PZI1(M1),PZIN(M1)
      LM=INT(0.5D0*DX1+1D0)
      IF (ITE.EQ.0) GOTO 125
      DO 400 I=2,M-1
```

```
      DO 210 J=2,N-1
      UXIJ=(U(I+1,J)-U(I-1,J))*DX1/2D0
      UZIJ=(U(I,J+1)-U(I,J-1))*DZ1/2D0
      VZIJ=(V(I,J+1)-V(I,J-1))*DZ1/2D0
      BC0=-UXIJ+(Z(J)*HX(I)*UZIJ-VZIJ)/H(I)
      PC=P(I,J)+BC0*CP*CP*DT
      P(I,J)=PC
  210 CONTINUE
  400 CONTINUE
c
c     I=1:
c
      CALL COEFF(W,A1,B1,1)
      DO 420 J=1,N
      IF (J.NE.N.OR.J.NE.1) THEN
      UXZ=(4D0*U(2,J+1)-3D0*U(1,J+1)-U(3,J+1))*DX1*DZ1/4D0
     1   -(4D0*U(2,J-1)-3D0*U(1,J-1)-U(3,J-1))*DX1*DZ1/4D0
      VXZ=(4D0*V(2,J+1)-3D0*V(1,J+1)-V(3,J+1))*DX1*DZ1/4D0
     1   -(4D0*V(2,J-1)-3D0*V(1,J-1)-V(3,J-1))*DX1*DZ1/4D0
      END IF
c
      IF (J.EQ.N) THEN
      UXZ=3D0*(4D0*U(2,N)-3D0*U(1,N)-U(3,N))*DX1*DZ1/4D0
     1   +(4D0*U(2,N-2)-3D0*U(1,N-2)-U(3,N-2))*DX1*DZ1/4D0
     1   -4D0*(4D0*U(2,N-1)-3D0*U(1,N-1)-U(3,N-1))*DX1*DZ1/4D0
      VXZ=3D0*(4D0*V(2,N)-3D0*V(1,N)-V(3,N))*DX1*DZ1/4D0
     1   +(4D0*V(2,N-2)-3D0*V(1,N-2)-V(3,N-2))*DX1*DZ1/4D0
     1   -4D0*(4D0*V(2,N-1)-3D0*V(1,N-1)-V(3,N-1))*DX1*DZ1/4D0
      END IF
      IF (J.EQ.1) THEN
      UXZ=4D0*(4D0*U(2,2)-3D0*U(1,2)-U(3,2))*DX1*DZ1/4D0
```

```
     1   -(4D0*U(2,3)-3D0*U(1,3)-U(3,3))*DX1*DZ1/4D0
     1   -3D0*(4D0*U(2,1)-3D0*U(1,1)-U(3,1))*DX1*DZ1/4D0
         VXZ=4D0*(4D0*V(2,2)-3D0*V(1,2)-V(3,2))*DX1*DZ1/4D0
     1   -(4D0*V(2,3)-3D0*V(1,3)-V(3,3))*DX1*DZ1/4D0
     1   -3D0*(4D0*V(2,1)-3D0*V(1,1)-V(3,1))*DX1*DZ1/4D0
         END IF
c
         VXX1J=(4D0*V(3,J)+2D0*V(1,J)-5D0*V(2,J)-V(4,J))*DX1*DX1
         VZIJ=W*(PZ1J(J)/H(1)+2D0*Z(J)*HX(1)*VXZ/H(1)-VXX1J)
         UXIJ=B1*U(3,J)+A1*U(2,J)-(A1+B1)*U(1,J)
     1      -W*(PX1J(J)-Z(J)*HX(1)*PZ1J(J)/H(1)
     1      +2D0*Z(J)*HX(1)*UXZ/H(1)
c
         BC0=-UXIJ-VZIJ
         PC=P(1,J)+BC0*CP*CP*DT
  420    P(1,J)=PC
c
c        I=M:
c
         CALL COEFF(W,A2,B2,-1)
         DO 430 J=1,N
         IF (J.NE.N.OR.J.NE.1) THEN
         UXZ=(3D0*U(M,J+1)+U(M-2,J+1)-4D0*U(M-1,J+1))*DX1*DZ1/4D0
     1    -(3D0*U(M,J-1)+U(M-2,J-1)-4D0*U(M-1,J-1))*DX1*DZ1/4D0
         VXZ=(3D0*V(M,J+1)+V(M-2,J+1)-4D0*V(M-1,J+1))*DX1*DZ1/4D0
     1    -(3D0*V(M,J-1)+V(M-2,J-1)-4D0*V(M-1,J-1))*DX1*DZ1/4D0
         END IF
         IF (J.EQ.N) THEN
         UXZ=3D0*(3D0*U(M,N)+U(M-2,N)-4D0*U(M-1,N))*DX1*DZ1/4D0
     1  +(3D0*U(M,N-2)+U(M-2,N-2)-4D0*U(M-1,N-2))*DX1*DZ1/4D0
     1  -4D0*(3D0*U(M,N-1)+U(M-2,N-1)-4D0*U(M-1,N-1))*DX1*DZ1/4D0
```

```
      VXZ=3D0*(3D0*V(M,N)+V(M-2,N)-4D0*V(M-1,N))*DX1*DZ1/4D0
     1  +(3D0*V(M,N-2)+V(M-2,N-2)-4D0*V(M-1,N-2))*DX1*DZ1/4D0
     1  -4D0*(3D0*V(M,N-1)+V(M-2,N-1)-4D0*V(M-1,N-1))*DX1*DZ1/4D0
      END IF
      IF (J.EQ.1) THEN
      UXZ=4D0*(3D0*U(M,2)+U(M-2,2)-4D0*U(M-1,2))*DX1*DZ1/4D0
     1  -(3D0*U(M,3)+U(M-2,3)-4D0*U(M-1,3))*DX1*DZ1/4D0
     1  -3D0*(3D0*U(M,1)+U(M-2,1)-4D0*U(M-1,1))*DX1*DZ1/4D0
      VXZ=4D0*(3D0*V(M,2)+V(M-2,2)-4D0*V(M-1,2))*DX1*DZ1/4D0
     1  -(3D0*V(M,3)+V(M-2,3)-4D0*V(M-1,3))*DX1*DZ1/4D0
     1  -3D0*(3D0*V(M,1)+V(M-2,1)-4D0*V(M-1,1))*DX1*DZ1/4D0
      END IF
C
      VXXMJ=(4D0*V(M-2,J)+2D0*V(M,J)-5D0*V(M-1,J)-V(M-3,J))*DX1*DX1
      VZIJ=W*(PZMJ(J)/H(M)+2D0*Z(J)*HX(M)*VXZ/H(M)-VXXMJ)
      UXIJ=B2*U(M-2,J)+A2*U(M-1,J)-(A2+B2)*U(M,J)
     1     +W*(PXMJ(J)-Z(J)*HX(M)*PZMJ(J)/H(M)
     1     +2D0*Z(J)*HX(M)*UXZ/H(M))
C
      BC0=-UXIJ-VZIJ
      PC=P(M,J)+BC0*CP*CP*DT
      P(M,J)=PC
  430 CONTINUE
C
C     J=1:
C
      DO 452 I=2,M-1
      UZZI1=(4D0*U(I,3)+2D0*U(I,1)-5D0*U(I,2)-U(I,4))*DZ1*DZ1
      IF (I.LT.LM) THEN
      UXIJ=-W*(PXI1(I)-UZZI1/HH(I))
     1     +B1*U(I+2,1)+A1*U(I+1,1)-(A1+B1)*U(I,1)
```

```
      END IF
      IF (I.GE.LM) THEN
      UXIJ=W*(PXI1(I)-UZZI1/HH(I))
     1     +B2*U(I-2,1)+A2*U(I-1,1)-(A2+B2)*U(I,1)
      END IF
      VZIJ=B1*V(I,3)+A1*V(I,2)-(A1+B1)*V(I,1)
     1     -W*H(I)*PZI1(I)
      BC0=-UXIJ-VZIJ/H(I)
      PC=P(I,1)+BC0*CP*CP*DT
      P(I,1)=PC
  452 CONTINUE
c
c     J=N:
c
      DO 450 I=2,M-1
      UXZ=(3D0*U(I+1,N)+U(I+1,N-2)-4D0*U(I+1,N-1))*DX1*DZ1/4D0
     1   -(3D0*U(I-1,N)+U(I-1,N-2)-4D0*U(I-1,N-1))*DX1*DZ1/4D0
      VXZ=(3D0*V(I+1,N)+V(I+1,N-2)-4D0*V(I+1,N-1))*DX1*DZ1/4D0
     1   -(3D0*V(I-1,N)+V(I-1,N-2)-4D0*V(I-1,N-1))*DX1*DZ1/4D0
      UZZIN=(4D0*U(I,N-2)+2D0*U(I,N)-5D0*U(I,N-1)-U(I,N-3))*DZ1*DZ1
      UZIN=(3D0*U(I,N)+U(I,N-2)-4D0*U(I,N-1))*DZ1/2D0
      VZIN=(3D0*V(I,N)+V(I,N-2)-4D0*V(I,N-1))*DZ1/2D0
      E1=(HXHX(I)+1D0)/HH(I)
      E2=2D0*HXHX(I)/HH(I)-HXX(I)/H(I)
      XMR=E1*UZZIN+E2*UZIN-2D0*HX(I)*UXZ/H(I)
      IF (I.LT.LM) THEN
      UXIJ=B1*U(I+2,N)+A1*U(I+1,N)-(A1+B1)*U(I,N)
     1     -W*(PXIN(I)-HX(I)*PZIN(I)/H(I)-XMR)
      END IF
      IF (I.GE.LM) THEN
      UXIJ=B2*U(I-2,N)+A2*U(I-1,N)-(A2+B2)*U(I,N)
```

```
     1      +W*(PXIN(I)-HX(I)*PZIN(I)/H(I)-XMR)
     c      UXIJ=(U(I+1,N)-U(I-1,N))*DX1/2D0
            END IF
            IF (I.LT.LM) THEN
            VXXIN=(4D0*V(I+2,N)+2D0*V(I,N)-5D0*V(I+1,N)-V(I+3,N))*DX1*DX1
            END IF
            IF (I.GE.LM) THEN
            VXXIN=(4D0*V(I-2,N)+2D0*V(I,N)-5D0*V(I-1,N)-V(I-3,N))*DX1*DX1
            END IF
            ZMR=VXXIN+E2*VZIN-2D0*HX(I)*VXZ/H(I)
            VZIJ=B2*V(I,N-2)+A2*V(I,N-1)-(A2+B2)*V(I,N)
     1      +W*(PZIN(I)/H(I)-ZMR)/E1
            UZIJ=UZIN
     c
            BC0=-UXIJ+(HX(I)*UZIJ-VZIJ)/H(I)
            PC=P(I,N)+BC0*CP*CP*DT
            P(I,N)=PC
     450    CONTINUE
     c
     c
     125    DO 123 J=1,N
            PX1J(J)=(4D0*P(2,J)-3D0*P(1,J)-P(3,J))*DX1/2D0
            PXMJ(J)=(3D0*P(M,J)+P(M-2,J)-4D0*P(M-1,J))*DX1/2D0
            IF (J.NE.N.AND.J.NE.1) THEN
            PZ1J(J)=(P(1,J+1)-P(1,J-1))*DZ1/2D0
            PZMJ(J)=(P(M,J+1)-P(M,J-1))*DZ1/2D0
            END IF
            IF (J.EQ.N) THEN
            PZ1J(J)=(3D0*P(1,N)+P(1,N-2)-4D0*P(1,N-1))*DZ1/2D0
            PZMJ(J)=(3D0*P(M,N)+P(M,N-2)-4D0*P(M,N-1))*DZ1/2D0
            END IF
```

```
      IF (J.EQ.1) THEN
      PZ1J(J)=(-3D0*P(1,1)-P(1,3)+4D0*P(1,2))*DZ1/2D0
      PZMJ(J)=(-3D0*P(M,1)-P(M,3)+4D0*P(M,2))*DZ1/2D0
      END IF
  123  CONTINUE
c
c
      DO 124 I=1,M
      PZI1(I)=(4D0*P(I,2)-3D0*P(I,1)-P(I,3))*DZ1/2D0
      PZIN(I)=(3D0*P(I,N)+P(I,N-2)-4D0*P(I,N-1))*DZ1/2D0
      IF (I.NE.1.AND.I.NE.M) THEN
      PXI1(I)=(P(I+1,1)-P(I-1,1))*DX1/2D0
      PXIN(I)=(P(I+1,N)-P(I-1,N))*DX1/2D0
      END IF
      IF (I.EQ.1) THEN
      PXI1(I)=(-3D0*P(1,1)-P(3,1)+4D0*P(2,1))*DX1/2D0
      PXIN(I)=(-3D0*P(1,N)-P(3,N)+4D0*P(2,N))*DX1/2D0
      END IF
      IF (I.EQ.M) THEN
      PXI1(I)=(3D0*P(M,1)+P(M-2,1)-4D0*P(M-1,1))*DX1/2D0
      PXIN(I)=(3D0*P(M,N)+P(M-2,N)-4D0*P(M-1,N))*DX1/2D0
      END IF
  124  CONTINUE
      RETURN
      END
     SUBROUTINE H0PS(H,PS,VM,HL,HR,DX,M)
     IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21,M1=21)
     DIMENSION H(M),A(N1),B(N1),C(N1),D(N1),
    1 E(N1),HX(N1),H0(N1),W(N1),G(N1)
   77  DO 57 I=1,M
```

```
57  H0(I)=H(I)
    PS0=PS
    DO 5 I=2,M-1
    HX(I)=(H(I+1)-H(I-1))/DX/2D0
 5  CONTINUE
    DO 10 I=2,M-1
    I1=I-1
    S0=HX(I)*HX(I)
    S1=SQRT(1D0+S0)
    S2=1.5D0*HX(I)*S1*PS*DX
    S3=S1*S1*S1
    A(I1)=-1-S2
    B(I1)=2D0
    C(I1)=-1+S2
    E(I1)=DX*DX*S3
    D(I1)=3*S0*PS*DX*DX*S1
10  G(I1)=DX
    D(1)=D(1)-A(1)*HL
    D(M-2)=D(M-2)-C(M-2)*HR
    B(M-1)=0D0
    D(M-1)=VM-DX*(HL+HR)/2D0
    CALL DIAG(A,B,C,E,G,D,W,M-2)
    DO 47 I=1,M-2
47  H(I+1)=W(I)
    H(1)=HL
    H(M)=HR
    PS=W(M-1)
    S=0D0
    DO 67 I=2,M-1
67  S=S+DABS(H0(I)-H(I))
    S=S+DABS(PS0-PS)
```

```
      WRITE(6,87)PS
   87 FORMAT(30X,'Ps=',E25.15/)
      IF (S.GE.5D-6) GOTO 77
      DO 75 I=1,M
      WRITE(6,'(6X,2HH=,F12.7)')H(I)
   75 CONTINUE
      RETURN
      END
      SUBROUTINE HORDER2(H1,M,N,CN)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21,M1=21)
      DIMENSION H1(M),A(N1),B(N1),C(N1),D(N1),
     1 E(N1),W(N1),G(N1)
      COMMON /U/U(M1,N1)/V/V(M1,N1)
     1 /H/H(M1),HH(M1)/HX/HX(M1),HXX(M1),HXHX(M1),
     1  HXXH(M1),HXH(M1)/P/P(M1,N1)
     1 /XZ/X(M1),Z(N1)/DXZ/DX,DZ
C
      H1(1)=0D0
      H1(M)=0D0
      DO 10 I=2,M-1
      I1=I-1
C
      UZ=(3D0*U(I,N)+U(I,N-2)-4D0*U(I,N-1))/DZ/2D0
      VZ=(3D0*V(I,N)+V(I,N-2)-4D0*V(I,N-1))/DZ/2D0
      UX=(U(I+1,N)-U(I-1,N))/DX/2D0
      VX=(V(I+1,N)-V(I-1,N))/DX/2D0
      UVXY=VZ-HX(I)*UZ+HX(I)*(-H(I)*VX+HX(I)*VZ+HX(I)*(UX*H(I)
     1   -HX(I)*UZ))
      S0=HX(I)*HX(I)
      S1=SQRT(1D0+S0)
```

```
      S2=-3D0*HX(I)*HXX(I)/S1
      S3=S1*S1*S1
      S4=-P(I,N)*S3+2D0*S1*UVXY/H(I)
      A(I1)=-1+0.5D0*S2
      B(I1)=2D0
      C(I1)=-1-0.5D0*S2
      E(I1)=DX*DX*S3
      D(I1)=-S4*DX*DX
   10 G(I1)=DX
      D(1)=D(1)-A(1)*H1(1)
      D(M-2)=D(M-2)-C(M-2)*H1(M)
      B(M-1)=0D0
      D(M-1)=0D0
      CALL DIAG(A,B,C,E,G,D,W,M-2)
      DO 47 I=1,M-2
   47 H1(I+1)=W(I)
      CN=W(M-1)
      RETURN
      END
      SUBROUTINE DIAG(A,B,C,H,G,D,U,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21)
      DIMENSION A(1:N+1),B(1:N+1),C(1:N+1),D(1:N+1),
     1 H(1:N+1),G(1:N+1),U(1:N+1),V(1:N1)
c
c     Note: we need (A(K),K=2,N),(B(K),K=1,N+1),(C(K),K=1,N-1),
c                (H(K),K=1,N),(G(K),K=1,N),(D(K),K=1,N+1)
c
      DO 12 K=2,N
      C1=-A(K)/B(K-1)
      B(K)=B(K)+C1*C(K-1)
```

```
      H(K)=H(K)+C1*H(K-1)
   12 D(K)=D(K)+C1*D(K-1)
C
      DO 22 K=1,N-1
      C2=-C(K)/B(K)
   22 G(K+1)=G(K+1)+C2*G(K)
      S1=0D0
      S2=0D0
      DO 32 K=1,N
      C3=G(K)/B(K)
      S1=S1+C3*H(K)
   32 S2=S2+C3*D(K)
      V(N+1)=(S2-D(N+1))/(S1-B(N+1))
      DO 42 K=1,N
   42 V(K)=(D(K)-H(K)*V(N+1))/B(K)
      U(N)=V(N)
      U(N+1)=V(N+1)
      DO 52 K1=1,N-1
      K=N-K1
      C4=-C(K)/B(K)
   52 U(K)=V(K)+C4*U(K+1)
      RETURN
      END
      SUBROUTINE THOMASX(D,B,E,FUV,J,M,N,UV)
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=21,M1=21)
      DIMENSION D(M),B(M),E(M),FUV(M),UV(M,N),
     1 R(N1),S(N1)
        MM1=M-1
        R(2)=FUV(2)/B(2)
        S(2)=-E(2)/B(2)
```

```
      DO 2543 K=3,MM1
      KM1=K-1
      RS=B(K)+D(K)*S(KM1)
      R(K)=(FUV(K)-D(K)*R(KM1))/RS
2543    S(K)=-E(K)/RS
      UV(M-1,J)=R(MM1)
      DO 2544 I=2,M-2
      K=M-I
2544    UV(K,J)=UV(K+1,J)*S(K)+R(K)
      RETURN
      END
    SUBROUTINE THOMASY(D,B,E,FUV,I,M,N,UV)
    IMPLICIT REAL*8 (A-H,O-Z)
    PARAMETER (N1=21,M1=21)
    DIMENSION D(N),B(N),E(N),FUV(N),UV(M,N),
   1  R(N1),S(N1)
      NM1=N-1
      R(2)=FUV(2)/B(2)
      S(2)=-E(2)/B(2)
      DO 2543 K=3,NM1
      KM1=K-1
      RS=B(K)+D(K)*S(KM1)
      R(K)=(FUV(K)-D(K)*R(KM1))/RS
2543    S(K)=-E(K)/RS
      UV(I,NM1)=R(NM1)
      DO 2544 I1=2,N-2
      K=N-I1
2544    UV(I,K)=UV(I,K+1)*S(K)+R(K)
      RETURN
      END
```

REFERENCE

1. B. J. Alfrink, *in* Numerical Methods in Laminar and Turbulent Flow, ed. C. Taylor and B.A. Schrefler, Proceedings of the Second International Conference, Venice, July 13-16, 1981, Pineridge Press.
2. G. K. Batchelor, An Introduction to Fluid Mechanics, Cambridge University Press, (1967)
3. E. B. Becker, G. F. Carey and J. T. Oden, Finite Elements, An Introduction, Prentice-Hall, Englewood Cliffs, NJ, 1 (1981)
4. J. B. Bell, P. Colella, and H.M. Glaz, *J. Comput. Phys.*, **85**, 257-283 (1989).
5. A. S. Benjamin and V. E. Denny, *J. Comput. Phys.* **33**, 340-358 (1979)
6. U. Bulgarelli, V. Casulli and D. Greenspan, On Numerical Treatment of free Surfaces for Incompressible Fluid-Flow Problems, *Nonlinear Anal Theory Methods Appl*, **4**, 975-983 (1984)
7. I-L Chern, J. Glimm, B. Lindquist, O. McBryan, B. Plohr and S. Yaniv, Front Tracking for Gas Dynamics, *J. Comput. Phys.*, **62**, 83-110 (1986)
8. A.J. Chorin, *J. Comput. Phys.*, **22**, 745-762 (1968)
9. A. J. Chorin, Flame Advection and Propagation Algorithms, *J. Comput. Phys.*, **35**, 1-11 (1980)
10. A.J. Chorin, *J. Comput. Phys.* **2**, 12-26 (1967)
11. J. Crank, Free and Moving Boundary Problem, Clarendon, Oxford, (1984)
12. C. Cuvelier and J. M. Driessen, Thermocapillary Free Boundaries in Crystal Growth, *J. Fluid Mech.*, **169**, 1-26 (1986)

13. J. T. Davies and E. K. Rideal, Interfacial Phenomena, Academic, New York, (1963)

14. S.C.R. Dennis and L. Quartapelle, *Int. J. Numer. Methods Fluids*, **9**, 871-890 (1989).

15. F. Dupret, A Method for the Computation of Viscous Flow by Finite Elements with Free Boundaries and Surface Tension, in Finite Element Flow Analysis, 4th Int. Symp., Tokyo, T. Kawai, Ed, North-Holland, Amasterdam, 495-502 (1982)

16. V. E. B. Dussan, On the Spreading of Liquid on Solid Surface: Static and Dynamic Contact Lines, *Ann Rev Fluid Mech* **11**, 371-400 (1979)

17. C. S. Ferderiksen and A. M. Watts, Finite Element Method for Time-Dependent Incompressible Free Surface Flow, *J. Comput. Phys.*, **39**, 282-304 (1981)

18. C. A. J. Fletcher, Computational Techniques for Fluid Dynamics, Springer-Verlag, Berlin, Heideberg (1988)

19. C. A. J. Fletcher, Computational Galerkin Methods, Springer-Verlag, New York, (1984)

20. J. M. Floryan and H. Rasmussen, Numerical Methods for Viscous Flows with Moving Boundaries, *Appl. Mech. Rev.*, **42**, No.12, 323-341 (1989)

21. M. Fortin, R. Peyret and R. Temam, *In* Proc. Second Int. Conf. Num. Meth. Fluid Dyn., ed. M. Holt, 337 Berlin (1971)

22. J. E. Fromm, A method for Computing Nonsteady Incompressible viscous fluid flows, Los Alamos Sci. Lab. Rep. LA-2910 (1963)

23. J. Glimm, B. Lindquist, O. McBryan, B. Plohr and S. Yaniv, Front Tracking Petroleum Reservoir Simulation, Proceedings of the Seventh SPE Symposium on Petroleum Reservoir Simulation, SPE-12238, 41-49, Society of Petroleum Engineers,Dallas (1983)

24. U. Ghia, K. N. Ghia and C. T. Shin, *J. Comput. Phys.* **48**, 387-411 (1982)

25. J.W. Goodrich and W.Y. Soh, *J. Comput. Phys.*, **84**, 207-241 (1989)

26. F. Harlow, Numerical Methods for Fluid Dynamics, An Annotated Bibliography, Los Alamos National Lab. Report LA-4281 (1969)

27. F. H. Harlow, A Machine Calculation Method for Hydrodynamic Problems, LAMS-1956, Los Alamos Scientific Lab., Los Alamos, New Maxico (1955)

28. F.H. Harlow and J.E. Welch, Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. Fluids* **8**, 2182-2189 (1965)

29. F.H. Harlow and J.E. Welch, *Phys. Fluids*, **8**, 2182-2189 (1965).

30. C. W. Hirt and B. D. Nichols, Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J. Comput., Phys.* **39**, 210-225 (1981)

31. H. Huang and H. Yang, The Computational Boundary Method for Solving the Navier-Stokes equations

32. J. M. Hyman, Numerical Methods for Tracking Interfaces, *Physica*, **12D**, 396-407 (1984)

33. J. M. Hyman and B. Larrouturou, The Numerical Differentiation of Discrete Functions Using Polynomial Interpolation Methods, *Appl. Math. and Comput.*, **10-11**, 487-506 (1982)

34. V. G. Jenson, *Proc. Roy. Soc. London Ser.* A **259**, 346 (1959)

35. I. S. Kang and L. G. Leal, Numerical Solution of Axisymmetric, Unsteady Free-Boundary Problems at Finite Reynolds Number, Part I: Finite Difference Scheme and Its Application to the Deformation of a Bubble in a Uniaxial Straining Flow, *Phys. Fluids* **30**, 1929-1940 (1987)

36. L. Kleiser and U. Schumann, in Proceedings, Third GAMM-Conference on Numerical Methods in Fluids Mechanics" (E.H. Hirschel, ED.), 165-173, Braunschweig, 1980.

37. J. Kim and P. Moin, *J. Comput. Phys.*, **59**, 308-323 (1985).

38. O. A. Ladyzhenskaya, The Mathematical Theory of Viscous, Incompressible Flow, 2nd, New York, Gordon & Breach (1969),Spring-Verlag (1971)

39. K. J. Laskey, Oran, E. S. and J. P. Boris, Approaches to Resolving and Tracking Interfaces and Discontinuities, Naval Research Lab. Report 5999, Arlington, VA (1987)

40. J. Lowndes, The Numerical Simulation of the Steady Movement of a Fluid Meniscus in a Capillary Tube, *J. Fluid Mech.*, **101**, 631-646 (1980)

41. Y-W Ma, *Mathematica Numerica Sinica*, **5**, (1983) in Chinese

42. H. Miyata T. Sato and N. Baba, Difference Solution of a Viscous Flow with Free-Surface Wave about an Advancing Ship, *J. Comput. Phys.*, **72**, 393-421 (1987)

43. R. E. Nickell, R. I. Tanner and B. Caswell, The Solution of Viscous Incompressible Jet and Free-Surface Flows Using Finite-Element Methods, *J. Fluid Mech.*, **65**, 189-206 (1974)

44. W. F. Noh and P. Woodward, SLIC (Simple Line Interface Calculation), in Lecture Notes in Physics, **59**, Springer-Verlag, New York, 330-340 (1976)

45. S. A. Orszag, *J. Fluid Mech.*, **49**, 75 (1971)

46. F. M. Orr and L. E. Scriven, Rimming Flow: Numerical Simulation of Steady, Viscous, Free-Surface Flow with Surface Tension, *J. Fluid Mech.*, **84**, 145-165 (1978)

47. S.V. Patankar and D.B. Spalding, Int. *J. Heat and Mass Transfer*, v.15, 1787 (1972)

48. R. B. Patne, *J. Fluid Mech.* **37**, 480 (1958)

49. D. W. Peaceman and H. H. Rachford, Jr., *J. Soc. Indust. Appl. Math.*, **3**, 28-41 (1955)

50. C. E. Pearson, *J. Fluid Mech.*, **21**. 611 (1965)

51. L. Quartapelle, *J. Comput. Phys.*, **40**, 453-477 (1981).

52. P.J. Roache, Computational Fluids Dynamics, Hermosa, Albuquerque, (1972).

53. G. Ryskin and L. G. Leal, Numerical Solution of Free-Boundary Problems in Fluid Mechanics, Part I: The Finite Difference Technique, *J. Fluid Mech.*, **148**, 1-17 (1984)

54. K. R. Reddy and R. I. Tanner, Finite Element Solution of Viscous Jet Flows with Surface Tension, *Comput. Fluids.*, **6**, 83-91 (1978)

55. K. J. Ruschak, A Method for Incorporating Free Boundaries with Surface Tension in Finite Element Fluid-Flow Simulators, *Int. J. Numer. Methods Eng.*, **148**, 1-17 (1980)

56. A. Rybicki and J. M. Floryan, "*Thermocapillary Convection in Liquid Bridges*", Phys. Fluids., **30**, No.7, pp. 1956-1972, 1987

57. R. Schreiber and H. B. Keller, *J. Comput. Phys.* **49**, 310-333 (1983)

58. W. J. Silliman and L. E. Scriven, Separating Flow near a Static Contact Line: Slip at a Well and Shape of a Free Surface, *J. Comput. Phys.*, ------- (1980)

59. A.B. Stephen, J.B. Bell, J.M. Solomon, and L.B. Hackerman, *J. Comput. Phys.* **53**, 152 (1984).

60. B. Swartz, In Mathematical Aspects of Finite Elements in Partial Differential Equations, Ed. C. de Boor, 279-312, Academic Press, New York (1974)

61. R. Temam, *Archiv. Ration. Mech. Anal.* **32**, 137-153 (1969a)

62. H. Yang, On Numerical Simulation of Navier-Stokes Equations Using Zero Perturbed Momentum and Divergence Boundary Conditions (In preparation)

63. H. Yang, H. Huang and B. R. Seymour, Computational Boundary Method for Two Dimensional Incompressible Flows (In preparation)

64. R. W. Yeung, Numerical Methods in Free Surface Flows, *Annu. Rev. Fluid Mech,* **14**, 395-442 (1982)

65. L.C. Woods, *Aeronautical Quarterly*, **5**, part 3, 176 (1954).

66. A. Thom, *Proc. Roy. Soc. London Ser.* **A 141**, 651 (1933)

67. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, The MAC Method: A Computing Technique For solving Viscous Incompressible, Transient Fluid-Flow Problems Involving Free Surface, Los Alamos Scientific Lab. Report LA-3425 (1966)