

DISSERTATION

AN IMPLICIT METHOD FOR WATER WAVE PROBLEMS

Submitted by

Martha B. Aston

Mathematics Department

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring, 1983

COLORADO STATE UNIVERSITY

December 9, 1982

WE HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER  
OUR SUPERVISION BY MARTHA B. ASTON  
ENTITLED AN IMPLICIT METHOD FOR WATER WAVE PROBLEMS

BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE  
DEGREE OF DOCTOR OF PHILOSOPHY

Committee on Graduate Work

	
_____	Department Head
_____	_____
_____	_____
_____	_____

## ABSTRACT

### AN IMPLICIT METHOD FOR WATER WAVE PROBLEMS

This paper presents an implicit scheme for numerically simulating fluid flow in the presence of a free surface. The scheme couples numerical generation of a boundary-fitted coordinate system with an efficient solution of the finite difference equations. The method solves the two dimensional Navier-Stokes equations by applying an implicit backward in time difference scheme which is linearized by Taylor series expansion about the known time level to produce a system of linear difference equations. The difference equations are solved by an Alternating-Direction-Implicit procedure which defines a sequence of one dimensional block tridiagonal matrix equations. A standard block elimination scheme solves the one dimensional equations. For each time step, solutions for all equations are calculated simultaneously and noniteratively. Preliminary solutions of free surface fluid flow in an open channel are presented. These solutions are examined to define initial stability criteria for the numerical scheme.

Martha Bell Aston  
Mathematics Department  
Colorado State University  
Fort Collins, Colorado 80523  
Spring, 1983

#### ACKNOWLEDGMENT

One of the greatest rewards I have had while working on this study has been to receive the encouragement of my friends and family, especially my mother. Their steadfast belief that this work would reach completion inspired me to continue to the end. Tom McCall, Steve Knox and Dean Wallace of the Colorado State University Computer Center understood the extensive computer resources necessary to perform the numerical computations and kindly helped in the debugging of the computer code. My advisor, Dr. J. W. Thomas, introduced me to numerical fluid dynamics and originally defined the problem addressed in this study. I am grateful for his patience during the months when my progress appeared to be slow.

My father, the late Dr. J. C. Bell, originally inspired in me a love for mathematics. At the beginning of this study, discussions with him were of great service to me. My husband, Dr. Graeme Aston, improved my understanding of the physical complexities of fluid flow and provided assistance in the preparation of this manuscript. He selflessly committed himself to my help the entire time I was working on this study. It is an honor for me to dedicate this paper to both of these gentlemen.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENT . . . . .	iii
LIST OF FIGURES. . . . .	v
<u>Chapter</u>	
INTRODUCTION . . . . .	1
I. BACKGROUND . . . . .	7
II. MATHEMATICAL FORMULATION . . . . .	16
III. NUMERICAL GRID GENERATION. . . . .	27
IV. NUMERICAL FORMULATION. . . . .	38
V. STABILITY ANALYSIS CONSIDERATIONS. . . . .	50
VI. APPLICATION TO PROBLEM . . . . .	53
VII. NUMERICAL RESULTS AND DISCUSSION . . . . .	69
REFERENCES . . . . .	78
APPENDIXES . . . . .	83

LIST OF FIGURES

	<u>Page</u>
1. Physical plane . . . . .	17
2. Free surface problem transformation. . . . .	29
3. Free surface elevation for three Froude numbers. . . . .	71
4. Numerical grid . . . . .	72
5. Movement of the surface of a bore. . . . .	76

## INTRODUCTION

The advent of high speed, large memory computers has encouraged the development of numerical techniques for solving complicated problems. In particular, problems in fluid dynamics have received much attention. This work presents a technique for numerically simulating the flow of a fluid in the presence of a free surface. The motion of a fluid with a free surface determines the dynamics of a wide variety of physical problems. Waves moving on a sloping beach, water flowing down a river and wave patterns created by a ship on or below the water surface represent just a few free surface fluid flow problems. The free surface is a fluid boundary on which pressure is prescribed but for which the position cannot be prescribed a priori. Furthermore, as the fluid moves the free surface may deform creating a time dependent and often complex geometry. Thus, in addition to calculating standard fluid dynamics parameters, numerical simulation of free surface flows must determine the unknown, time dependent domain defined by the moving free surface.

The specific free surface problems to which the numerical scheme presented here applies involve the motion of a viscous, incompressible fluid. Atmospheric pressure is imposed on the free surface and a uniform body force,

gravity, acts on the fluid. In addition to the free surface, other types of fluid boundaries, such as fixed walls or lines of uniform flow, may be present. As presented here, the scheme solves two dimensional flow problems and allows for ready extension to three dimensional flows.

The equations governing the motion of an incompressible, viscous fluid form a system of nonlinear, coupled, multidimensional partial differential equations. Any reasonably tractable mathematical model of fluid flow is bound to contain some simplification. Even so, numerical simulation of viscous flows require large amounts of computer time and storage. It is also very often difficult to correctly apply boundary conditions, especially for fluids with a free surface. Such difficulties complicate the formulation of numerical methods for solving fluid dynamics problems.

The solution technique presented here uses finite difference approximations to the system of governing equations. Hence, the numerical solution is found by solving a system of algebraic equations which are defined at grid points determined by the grid spacings,  $\Delta x$  and  $\Delta y$ . For time dependent problems, such as presented in this paper, the solution is evaluated at a sequence of time steps, of spacing  $\Delta t$ , with unknown values at the advanced time level being determined from values of the variables at a few adjacent time levels. Several currently established

methods for solving free surface flow problems use explicit finite differencing schemes. Although the algebraic system of an explicit scheme is easily solved (each algebraic equation involves only one unknown value), such schemes are subject to stability criteria which restrict the size of the time step relative to the spatial mesh size. Stability restrictions of explicit solution schemes determine the maximum time increment from the spatial mesh size, not from the rate at which physical variables are changing with time. These computational restrictions can be disadvantageous in the numerical solution of fluid dynamics problems. For example, suppose a solution is sought for a largely inviscid flow which has a thin viscous boundary layer in which the solution must be resolved by a finer mesh than is desirable to use in the inviscid region. Since the time step is usually required to be proportional to the smallest mesh spacing in the domain, many time steps may be required to achieve small changes in the flow parameters in the inviscid region. Similarly, if an explicit finite difference scheme is used to compute a steady state flow as the limit of a time dependent problem, the time step may not be increased when the solution is changing slowly with time and steady conditions are approached. Thus, explicit scheme stability criteria can lower computational efficiency by reducing the time steps with which the solution is advanced through time.

To avoid the stability restrictions described above, the numerical method of this paper implements an implicit finite difference scheme. Implicit differencing defines an algebraic system in which each equation contains unknown values from more than one grid point at the advanced time level. The resulting algebraic system is generally more difficult to solve than the explicitly differenced algebraic system. However, implicit schemes tend to be stable for large time steps and limits on the size of the time step are more closely tied to changes in the physical variables. The computational efficiency of implicit finite difference schemes is primarily dependent on the efficient solution of the resulting algebraic system. The solution scheme presented in this paper attains computational efficiency by exploiting the structure of the algebraic system and by employing an Alternating-Direction-Implicit (ADI) solution for the system of finite difference equations.

For many types of differential systems, boundary conditions have a strong influence on the character of the solution. In the solution of free surface problems, the free surface boundary conditions often have the most noticeable effect on the dynamics of the fluid flow. This influence makes careful incorporation of boundary conditions essential in the finite difference solution of fluid flow equations. Indeed, poor numerical representation of boundary conditions compromises the accuracy of the

numerical scheme, nullifying any benefits of computational efficiency. A conceptually straightforward way to promote accuracy is to construct a computational grid with the property that each physical boundary is coincident with a computational grid line. In this manner, finite difference expressions can be calculated using only values at the computational grid points without the need for interpolation between grid points. Such a finite difference computational grid is advantageous in the numerical simulation of free surface fluid flows which are characterized by nonlinear boundary conditions. The free surface boundary conditions require knowledge of both the location and the slope of the surface. Hence, avoiding interpolation to evaluate free surface parameters is particularly important. However, for free surface problems, this finite difference grid must be constructed to fit the boundaries of a time dependent domain which is defined by the free surface boundary conditions themselves. Again, note that because the physical domain is time dependent and because its location can be found only by solving the flow and boundary differential equations, determinations of a computational domain must be made at each time step.

The solution scheme presented in this paper solves the free surface fluid flow equations and determines the computational domain as the solution of a system of two dimensional parabolic partial differential equations which are defined on the physical domain. The system of flow

equations and all boundary conditions are transformed from the physical domain to a computational domain; boundaries of the physical domain are mapped to boundaries of the computational domain. Finally, the entire system of fluid flow and grid generating equations is solved at each time step in the computational domain.

CHAPTER I  
BACKGROUND

The nonlinearities which characterize many of the interesting properties of fluid motion make the equations of fluid dynamics difficult to solve by analytic techniques, except in a variety of special cases. Thus, the majority of fluid dynamics problems still defy solution by classical analysis and must be solved by numerical methods. Although most numerical schemes for simulating fluid flows incorporate the principle of conservation of fluid mass and momentum, many factors, such as boundary conditions, problem geometry and fluid properties, influence and complicate the development of these numerical solutions. As with other types of fluid dynamics problems, many workers have been challenged by the numerical simulation of free surface flows. Von Kerczek [1] gives an informative review of numerical methods for free surface problems. Roach [2] also carefully describes several numerical techniques currently applied to free surface problems. The intent of this section is to illustrate the variety of solution methods applicable to free surface problems and to discuss schemes which affected the development of the method presented in this paper.

To simplify the physical problem, it is frequently assumed that the fluid is inviscid and irrotational. Then, for an incompressible fluid in a simply connected region, there exists a velocity potential which satisfies the Laplace equation. Even though the incompressible potential flow is highly specialized, it does provide insight into more complicated flows. Despite these simplifying assumptions, which reduce the fluid flow equations to one simple equation, the nonlinear free surface boundary conditions make straightforward analysis difficult.

One promising technique for numerically solving the potential flow free surface problem involves the use of spectral methods. Spectral methods expand the velocity potential in a finite number of terms from a complete set of functions which satisfy the Laplace equation. The unknown expansion coefficients are then frequently specified by a boundary dependent functional, for example minimization of the error caused by approximating the free surface boundary condition with the specified functions. Haussling and Van Eseltine [3] demonstrate that the spectral method can be applied successfully to potential flow problems. Their work also points out some of the limitations of this method. First, it is difficult to define boundary conditions for finite domains (nontrivial potential flow is defined on infinite domains) in such a way that instabilities do not arise. Secondly, extension to

multiply connected domains or to three dimensional domains is computationally and conceptually difficult. Haussling and Van Eseltine suggest that methods more adaptive to the complex geometries characteristic of free surface flows should also be explored.

Fritts et al. [4] examine incompressible, inviscid, irrotational flow with the intent of determining the effect of ocean waves on submerged or partially submerged structures, situations which quickly define complex geometries. These researchers implement a Lagrangian finite difference method. Lagrangian calculations utilize a computational mesh composed of points embedded in the fluid and moving with it. Thus, the fluid initially in the interior of a finite difference cell always remains in that cell and the fluid boundaries, notably the free surface, always move with the computational cell boundaries. Simple and accurate treatment of the free surface boundary conditions follows, as is essential because the free surface dominates the entire flow. However, as Fritts points out, the strength of the Lagrangian method is also its weakness since the mesh often becomes very deformed. In such situations, the finite difference approximations used by Fritts et al. lose accuracy. Cell distortion can be particularly bad when shear flows, characteristic of viscous fluids near solid boundaries, or large amplitude water waves occur. Nonetheless Fritts et al. do obtain favorable results when

applying Lagrangian methods to flows which do not undergo large distortions.

More recently, McCormick and Thomas [5] have developed an iterative scheme incorporating multigrid methods in the solution of the two dimensional potential flow problem in the presence of a free surface. This iterative process alternately improves the stream function values and then the free surface boundary values. Although their method is presented in an introductory form and results are preliminary, the multigrid scheme promises to be an efficient iterative method for solving the elliptic velocity potential problem. These authors note that, as with most iterative solutions of nonlinear problems, computational efficiency and indeed even success depend on a good initial guess of the flow parameters. One disadvantage common to all methods which solve the potential problem is the necessity of using numerical differentiation to obtain fluid velocities which increases the possibility of numerical error.<sup>1</sup> Similarly, with these methods, pressures are not determined directly in the velocity potential formulation and so must be obtained by separate calculations.

Equations defining the full viscous flows are considerably more complicated than the Laplace equation which defines potential flow. Viscous free surface flows are not

---

<sup>1</sup> Fritts et al. solve for the Lagrangian fluid velocities and so must perform numerical integration to determine the actual fluid velocities.

determined by the stream function and the free surface alone, fluid vorticity must also be calculated. Thus, in addition to the elliptic (in this case Poisson) stream function equation, it is necessary to solve a parabolic nonlinear vorticity equation. When vorticity and the stream function are chosen as independent variables, free surface boundary conditions are awkward. Hence, for free surface viscous flows it is customary to use a primitive variable formulation where velocity and pressure define the independent variables. Use of primitive variables not only simplifies the expression of free surface boundary conditions but also gives direct calculation of the fluid velocities and transient pressures.

Finite difference techniques have proven successful in solving the primitive variable formulation of the free surface viscous fluid dynamics problem. One of the most highly refined finite difference algorithms for free surface flows is the Marker and Cell (MAC) scheme originated by Harlow and Welch [6]. The MAC scheme solves the full Navier-Stokes equations for a viscous, incompressible fluid. Finite differences are calculated on a fixed (Eulerian) mesh superimposed over the fluid. A set of marker particles, which have no physical meaning, are carried along in the calculations to keep track of the moving free surface. Calculations are performed in several steps:

- a. The pressure is determined by solving a finite difference Poisson's equation whose source term is a function of the velocities. This equation is subject to the requirement that the resulting momentum equations produce a velocity field that satisfies the incompressibility condition.
- b. Explicit finite difference Navier-Stokes equations are solved to find the fluid velocities.
- c. Marker particles are moved in accordance with the fluid velocities and so trace the movement of the free surface.

Free surface boundary conditions are applied at the center of the mesh cell containing the fluid interface and no attempt is made to determine the exact location of the free surface within the calculational cell.

Many investigators have worked to improve the representation of free surface boundary conditions, marker particle bookkeeping efficiency and the applicability to three dimensions for the original scheme. Chan and Street [7] offer a noteworthy improvement by devising a way to apply free surface boundary conditions at the actual location of the free surface instead of relying on linear interpolation to evaluate free surface parameters. Nichols and Hirt [8,9] have effectively incorporated several MAC refinements in their efficient three dimensional algorithm, SOLA-SURF. Numerical results from such MAC type programs are convincing and challenge other workers who attempt to

numerically simulate free surface flows. It should be noted, however, that all MAC type procedures depend on a somewhat artificial treatment of boundary conditions. In MAC calculations, the finite difference expressions assign values to velocity components at mesh points outside of the fluid<sup>2</sup> in a situation where no physical principle describes fluid velocities outside the fluid domain. Also, ultimate computational efficiency is governed by the stability restrictions of the explicit differencing scheme. Even utilizing the maximum allowable time step to advance velocities does not necessarily maximize computational efficiency since larger time steps only increase the number of iterations necessary to solve the Poisson pressure equation.

The numerical method presented in this paper relies on numerical generation of a computational mesh which conforms to the time dependent fluid domain determined by the free surface. Several techniques applicable to free surface problems have been developed for numerical grid generation. This paper utilizes work which has been done at Mississippi State University [10-12]. Workers there are developing a grid generating technique adaptable to situations in which the fluid domain has a complicated, time dependent

---

<sup>2</sup> A fixed mesh covers the moving fluid and so it is necessary to define computational points at all locations where fluid may appear. Also, velocities and pressure are calculated at staggered points which necessitates defining flow parameters outside the fluid domain.

geometry. The solution procedure maps the physical domain onto a computational domain. The coordinates of the computational domain are determined by elliptic mapping equations. To maximize accuracy within the physical domain, computational grid points may be clustered by manipulation of boundary conditions and forcing functions for the mapping equations. Thames et al. [13] demonstrate the application of the computational grid generating scheme to fluid flows in multiply connected regions (i.e., the domains occurring in flow around several bodies). Shanks [14] applies the method to time dependent free surface flow problems by numerically determining the computational grid and then solving the flow equations in separate calculations. These works provide effective grid generation and hence an accurate means of numerically representing boundary conditions. They do not attempt to couple computational grid generation with an efficient numerical solution to fluid dynamics problems.

The scheme presented here defines a method for integrating effective computational grid generation with an efficient solution of the primitive variable form of the free surface flow equations. In addition, the present method includes simultaneous solution of fluid flow, free surface boundary conditions and grid generation equations and so should give some indication as to whether such a procedure is computationally advisable. The numerical scheme of this work also uses no iteration to advance all

flow variables one time step. To exploit favorable stability properties and to increase computational efficiency, the numerical solution scheme presented here uses full implicit backward differencing coupled with an ADI matrix solver. This scheme is described by Aston and Thomas [15].

The aim of this present work is to investigate the feasibility of combining the above described solution components into one numerical scheme. Consequently, much effort has been expended on correctly coupling the various elements of the scheme to produce a reasonable system of equations which may be solved without the imposition of restrictive stability criteria. This emphasis, plus the limited computational resources available to the author, has resulted in only preliminary testing to date of a few simple fluid flow geometries. Also, the scheme currently includes no capability to cluster computational grid lines; however, such capability could be incorporated in the grid generating equations. Results concerning stability and parameter values are primarily due to numerical experimentation.

## CHAPTER II

### MATHEMATICAL FORMULATION

The equations which describe the simple physical situation that occurs when an incompressible viscous fluid flows through a channel with a rough bottom exhibit the complexity of numerically simulating free surface fluid flows. Figure 1 illustrates schematically such a geometry. A coordinate system is introduced in which the y-axis is directed opposite to the force of gravity and the x-axis parallels undisturbed portions of the free surface. The free and bottom surfaces are denoted by  $y = H(x,t)$  and  $y = F(x)$ , respectively. Consider a fluid of undisturbed depth,  $d$ , occupying the region,  $D_f$ , of length,  $l$ , and let the fluid move from left to right with a velocity  $\vec{u} = u\vec{i} + v\vec{j}$ . The two dimensional Navier-Stokes equations govern the motion of the fluid through this region by prescribing continuity of fluid momentum. Continuity of fluid mass is imposed by the divergence equation. Thus, written in conservation form, the equations to be solved are

$$(1) \quad \vec{u}_t + (\vec{u} \cdot \nabla) \vec{u} = - \nabla \left( \frac{p}{\rho} \right) + \nu \nabla^2 \vec{u} + \vec{E} \quad \text{in } D_f$$

$$(2) \quad \nabla \cdot \vec{u} = 0 \quad \text{in } D_f$$

where  $p$  denotes the pressure,  $\rho$  denotes the fluid density,  $\nu$  is the kinematic viscosity and  $\vec{E}$  is the external force (specifically,  $\vec{E}$  contains the gravity term so that here

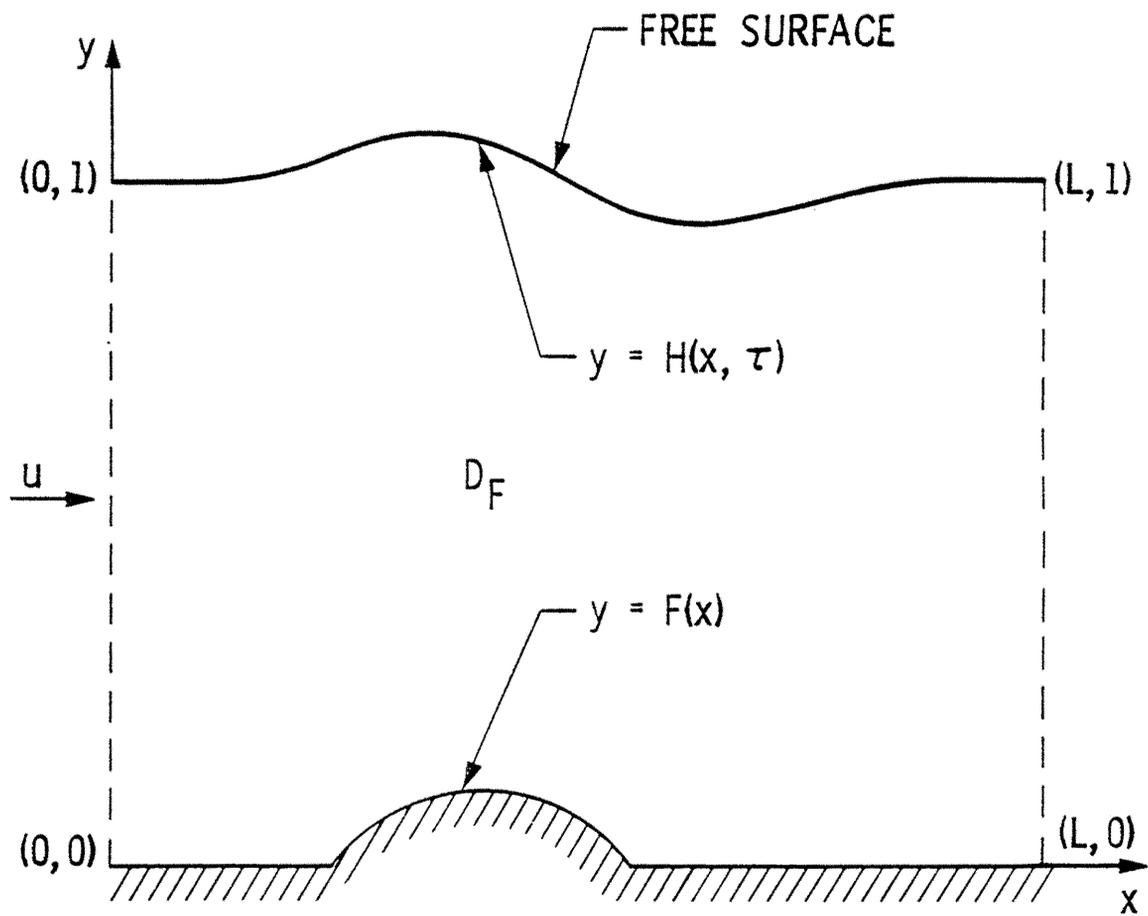


Fig. 1 Physical plane

$\vec{E} = -g\vec{j}$ ). For most problems describing water (or liquid) flow, incompressibility is a reasonable assumption, expressing the fact that the pressure depends so strongly on the density that small changes in the latter produce very large changes in the former. Hence, take  $\rho$  to be constant. The term  $\nu$  represents molecular viscosity only and is assigned a constant value. These equations assume non-turbulent flow and neglect any effects of temperature variation. Note also that by specifying three components for the velocity,  $\vec{u}$ , equations (1) and (2) describe three dimensional fluid flow.

For a fluid entering the left hand side (inflow boundary) of  $D_f$  with velocity  $\vec{u} = u_0\vec{i}$ , the flow is characterized with the following nondimensional variables denoted by primes:

$$\begin{aligned} x' &= x/d & y' &= y/d \\ u' &= u/u_0 & p' &= p/(\rho u_0^2) \\ \tau &= \tau u_0/d & Re &= u_0 d/\nu \\ Fr &= u_0/\sqrt{gd} & l' &= l/d \end{aligned}$$

where the Reynolds number,  $Re$ , measures the ratio of the inertia to the viscous force and the Froude number,  $Fr$ , measures the ratio of inertia to the gravity force.

Expressing equations (1) and (2) in terms of the non-dimensional variables with the primes dropped yields

$$(3) \quad \vec{u}_\tau + (\vec{u} \cdot \nabla)\vec{u} = -\nabla p + (1/Re)\nabla^2\vec{u} + \vec{E} \quad \text{in } D_f$$

$$(4) \quad \nabla \cdot \vec{u} = 0 \quad \text{in } D_f$$

where here  $\vec{E} = -(1/Fr^2)\vec{j}$ . Equation (3) defines the local change in momentum,  $u_\tau$ ,<sup>1</sup> in terms of the convection of momentum by fluid motion,  $(\vec{u} \cdot \nabla)\vec{u}$ , and the momentum change resulting from normal pressure forces,  $-\nabla p$ ; the term  $(1/Re)\nabla^2\vec{u}$  represents the diffusion of momentum by viscous forces and the last term,  $\vec{E}$ , describes momentum production by gravity. For an incompressible fluid, equation (4), often called the equation of continuity, expresses the fact that locally there can be no net increase or decrease in the fluid mass.

Two aspects of these equations merit close attention. First, the value of the Froude number has a decided effect on the movement of the free surface (see Stoker [16]). In general, decreasing the Froude number results in a lowering of wave amplitude on the free surface. For special linearized channel flows, for example very slow flow of an inviscid fluid in a situation where surface waves are long compared with the depth of the fluid, analytic relations may be determined which describe the depth of the water ( $H(x,t) - F(x)$  in Figure 1) in terms of the Froude number. In this idealized case, a Froude number greater than one specifies that the free surface changes height in the same way the bottom changes height; for a Froude number less than one, the free surface height decreases when the channel bottom is elevated. It would be overly optimistic

---

<sup>1</sup> Note that equation (1), and hence equation (3), has been normalized with respect to fluid density,  $\rho$ .

to expect the solution of the current nonlinear problem to respond in direct accordance with this result, indeed, it does not. However, the linearized theory does emphasize the importance of the value of the Froude number in the numerical simulation of free surface flows.

The second notable feature of the equations that determine the fluid dynamics is the inherent difference between equations (3) and (4). Equation (3) is parabolic in time; it poses an initial value problem in which the solution is advanced through time from some initial condition. Equation (4) poses a boundary value problem and is usually solved by an iterative numerical scheme. One of the central challenges of computational fluid dynamics lies in the pairing of these two different types of partial differential equations. Unfortunately, a numerical scheme designed to accurately and efficiently solve the parabolic equation may not be compatible with accurate and efficient solution of the continuity equation. Furthermore, these equations are linked by flow parameters on the boundary of the fluid domain. Resolution of this numerical inconvenience is basic to any numerical technique for simulating fluid flows.

The numerical method of this paper uses an ADI finite difference scheme to solve equation (3) which advances the velocity field in time. However, equations (3) and (4) suggest no obvious means for advancing the pressure in time. Introducing an artificial compressibility term into

the continuity equation, makes it possible to evaluate the pressure in a natural and efficient way. A modification of equation (4), the equation of artificial compressibility is

$$(5) \quad \beta p_{\tau} = - \nabla \cdot u$$

where  $\beta \ll 1$  is the coefficient of artificial compressibility. Note that for problems seeking a steady state solution, the term  $\beta p_{\tau}$  vanishes as steady state is approached and equation (5) reduces to equation (4).

Inclusion of the term  $\beta p_{\tau}$  adapts the equation of continuity to the ADI scheme applied to equation (3). The scheme presented here solves equations (3) and (5) in the fluid domain,  $D_f$ .

Other workers, notably Chorin [17-18] and Steger and Kutler [19], have used the concept of artificial compressibility to successfully simulate incompressible fluid flows without free surfaces. For a specified initial boundary value (only fixed boundaries are considered), Teman [20] analytically proves the solution of equations (3) and (5) converges to the solution of equations (3) and (4) as  $\beta \rightarrow 0$ . Teman also demonstrates that for a given discretization of the specified artificial compressibility problem, the discrete approximations converge to the solution of equations (3) and (4) as  $\beta \rightarrow 0$  and the computational mesh is refined. Such results, though not directly applicable to the free surface problem addressed in this paper, are encouraging.

To gain physical insight on the effect of including the artificial compressibility term, consider the following simple equation of state for the fluid:

$$(6) \quad p = A\rho^\delta$$

where  $A$  is constant and  $\delta$  is the ratio of specific heats at constant pressure and constant volume. Equation (6) describes a known relation between pressure and density in an inviscid fluid. Substitution of equation (6) into equation (5) gives

$$(7) \quad \beta A \delta \rho^{\delta-1} \rho_\tau = -\nabla \cdot \vec{u}$$

or equivalently

$$(8) \quad \beta \delta \left(\frac{p}{\rho}\right) \rho_\tau = -\nabla \cdot \vec{u}.$$

The speed of sound in the inviscid fluid,  $c$ , is given by

$$(9) \quad c = \sqrt{\delta p / \rho}$$

thus equation (8) may be expressed

$$(10) \quad \beta c^2 \rho_\tau = -\nabla \cdot \vec{u}.$$

Associated with each compressible fluid is a modulus of compression which measures the compressibility of the fluid (see Shapiro [21]). Taking  $\beta$  to be the reciprocal of this modulus, set

$$(11) \quad \beta = \frac{1}{\rho c^2}$$

and so equation (10) becomes

$$(12) \quad \rho_\tau = -\rho \nabla \cdot \vec{u}$$

Finally, if  $|\rho_x| \ll 1$  and  $|\rho_y| \ll 1$ , which is true for fluids which are essentially incompressible, then equation (12) is approximately

$$(13) \quad \rho_\tau = -\nabla \cdot (\rho \vec{u})$$

which defines the conservation of mass for a compressible fluid. Thus equation (5) does approximate conservation of mass for a slightly compressible fluid. Also, note that the modulus of compression,  $1/\beta$ , increases as the fluid becomes increasingly incompressible. Hence, taking  $\beta$  small makes equation (5) approximate an incompressible fluid.

The most influential boundary associated with free surface problems is the free surface,  $y = H(x, \tau)$ , itself. Free surface boundary conditions are formulated from the following principles:

- a. Tangential stress vanishes at the free surface.
- b. On the free surface, normal stress balances the externally applied normal stress.
- c. Change in the surface elevation is determined by the local fluid velocity.

The first principle implies that the atmosphere cannot exert a shear stress; that is, the viscous stresses of the atmosphere are assumed to be negligible. Surface tension is also neglected. To express these principles mathematically, first the  $x$  and  $y$  components of the unit outward normal to the free surface are calculated. These  $x$  and  $y$  components are respectively

$$(14) \quad n_x = -H_x [1 + (H_x)^2]^{-1/2}$$

$$(15) \quad n_y = [1 + (H_x)^2]^{-1/2} .$$

The corresponding cartesian components of the unit tangential vector are

$$(16) \quad m_x = [1 + (H_x)^2]^{-1/2}$$

$$(17) \quad m_y = H_x [1 + (H_x)^2]^{-1/2} .$$

Applying the first two principles to assure a balance of stresses at the free surface results in the following equations:

$$(18) \quad (1/Re)\{2u_x H_x - [1 - (H_x)^2][u_y + v_x] - 2H_x v_y\} = 0$$

on  $y = H(x, \tau)$

$$(19) \quad (1/Re)\{2u_x (H_x)^2 - 2H_x (u_y + v_x) + 2v_y\}$$

$= (p - p_0) [1 + (H_x)^2] \quad \text{on } y = H(x, \tau)$

where  $p$  is the water pressure on the free surface and  $p_0$  is the atmospheric pressure. The third principle is applied to determine the change in the free surface height,  $H_\tau$ , from the vertical component of the fluid motion plus the horizontal convection of the free surface elevation:

$$(20) \quad H_\tau + u H_x - v = 0 \quad \text{on } y = H(x, \tau) .$$

Equations (18) to (20) define the free surface boundary conditions.

Along the bottom of the flow domain,  $y = F(x)$ , the fluid velocity is determined by prescribing free slip boundary conditions; a no slip boundary could be imposed if desired. Accordingly, set

$$(21) \quad \vec{n} \cdot \vec{u} = 0 \quad \text{on } y = F(x)$$

$$(22) \quad \nabla u_\tau \cdot \vec{n} = 0 \quad \text{on } y = F(x)$$

where  $\vec{n}$  is the inward pointing normal at the boundary and  $u_\tau$  is the velocity component tangential to  $y = F(x)$ . On  $y = F(x)$  an equation defining pressure may be derived from

the momentum equations as shown in Appendix A. At the inflow and outflow boundaries a uniform flow with hydrostatic pressure is assumed. For computational requirements, the inflow and outflow conditions are set at finite values of the channel length which are sufficiently removed from irregularities in the bottom surface so that uniform flow at these boundaries is a reasonable approximation to physical reality. The equations describing this uniform flow at the inflow and outflow boundaries take the form

$$(23) \quad \vec{n} \cdot \vec{u} = u_0 \quad \text{on } x = 0, x = 1$$

$$(24) \quad p = p_0 + [H(x, \tau) - y]/Fr^2 \quad \text{on } x = 0, x = 1$$

Here  $\vec{n}$  is the unit normal to the boundary in the direction of flow and  $Fr$  is the Froude number. The boundary conditions outlined above provide an adequate physical model to permit testing of the numerical scheme presented here without unnecessarily complicated calculations. However, the computational technique of this paper is independent of these specified boundary conditions and ultimately other types of boundary conditions could be incorporated in the implementation of this scheme.

The goal of the mathematical scheme presented here is to find the solution to equations (3), (4) and (18) to (20). This solution is sought for the boundary conditions defined by equations (21) to (24) and for the initial conditions outlined in Chapter IV. As suggested by the results of Teman [20], the solution is determined by

numerically solving equations (3), (5) and (18) to (20) for small values of the coefficient of artificial compressibility,  $\beta$ .

## CHAPTER III

### NUMERICAL GRID GENERATION

Any effort to numerically solve the above described equations must successfully handle the computational difficulties posed by the time dependent fluid domain. A finite difference solution for a free surface problem calculated by use of a fixed grid imposed on the moving, often geometrically complicated, domain almost certainly requires some evaluation of boundary values by interpolation among values at grid points. As found by the developers of the MAC schemes, such interpolation on boundaries can destabilize a numerical solution that otherwise could be useful. To avoid this difficulty, the solution scheme of this paper numerically maps the physical domain into a fixed computational domain with time dependent functions. The mapping is specified so that the boundaries of the computational domain are coincident with the boundaries of the physical plane. The equations defining the fluid dynamics are transformed (mapped) into the computational domain where the numerical solution is computed.

The basic grid generating technique employed here receives inspiration from the usefulness of conformal mappings defined on the complex plane. However, because the scheme of this paper must be adaptive to complicated geometries and three dimensional problems, it is not

desirable to restrict attention to conformal mappings. More generally, partial differential equations are sought which can be used to produce a numerical mapping from a finite difference grid in the physical plane to a fixed grid in the computational plane. For problems with fixed, geometrically complicated physical domains, Steger [22] and Thames [10] demonstrate the usefulness of such a numerical mapping in the simulation of viscous flows. In the present application, the image of these fixed computational grid lines are lines in the physical domain which, though moving in time, should not cross or coalesce. The time dependent mapping from the physical to the computational plane must be one-to-one and map boundaries to boundaries. Though the grid in the computational domain is orthogonal, the grid determined by the inverse mapping in the physical domain need not be orthogonal. In fact, computational experience of other investigators, for example Shanks and Thompson [11], shows that such orthogonality is not required.

A mapping of this type, for the physical region  $D_f$  illustrated in Figure 1 into a fixed rectangle in the computational plane, is shown in Figure 2. The free surface, AB, maps onto the bottom of and the sides, AD and BC, map onto the left and right sides of , respectively. For more complex geometries, such as multiply connected regions, the mapping will be more complex. However, for the preliminary results presented here, the above described

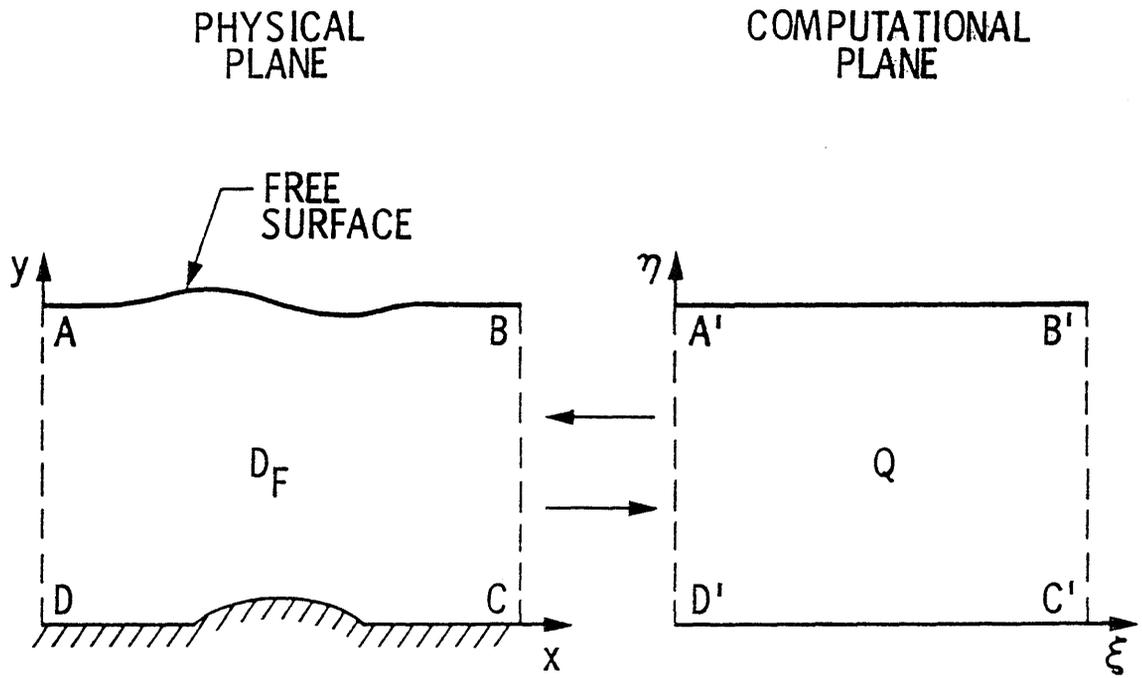


Fig. 2 Free surface problem transformation

configuration and mapping is considered sufficient to test overall feasibility of the proposed solution scheme.

The transformation from the nondimensional physical plane  $(x,y)$  to the computational plane  $(\xi,\eta)$  is defined by the mapping functions,  $\xi = \xi(x,y,\tau)$  and  $\eta = \eta(x,y,\tau)$  (and for completeness,  $\tau = \tau$ ). The variables  $\xi$  and  $\eta$  are chosen to satisfy the following partial differential equations:

$$(25) \quad \epsilon \xi_{\tau} - \nabla^2 \xi = 0 \quad \text{in } D_f$$

$$(26) \quad \epsilon \eta_{\tau} - \nabla^2 \eta = 0 \quad \text{in } D_f$$

where  $\epsilon > 0$  is specified. Thus, in addition to solving the flow equations described previously, equations (25) and (26) are solved to determine the domain in which calculations will be performed. Other authors, for example Warsi and Thompson [12], Chu [23] and Haussling and Coleman [24], frequently determine the mapping with elliptic equations and then control the spacing of the computational coordinates by requiring that  $\xi$  and  $\eta$  satisfy a Poisson equation. Final implementation of the present scheme could incorporate this type of modification. In fact, the inclusion of coordinate placement capabilities is probably necessary before the scheme presented here can be applied successfully to more complicated geometries or to viscous boundary layer flows.

The difference between computational grid generating equations (25) and (26) and other more common grid generating equations is the addition of the  $\epsilon \xi_{\tau}$  and  $\epsilon \eta_{\tau}$  terms. However, it should be noted that for flows which

approach a steady state, these terms become small and equations (25) and (26) approach the usual mapping equations of other investigators. A more fundamental difference exists between the grid generation employed here and standard grid generation techniques. Usual practice determines the computational domain and then in a separate calculation the flow equations are transformed to the computational domain and solved; in this paper it is proposed to determine the computational domain and solve the flow equations in the computational domain in one step. Thus, grid generation and flow equations are solved simultaneously at each time step. For time dependent problems the terms  $\epsilon \xi_{\tau}$  and  $\epsilon \eta_{\tau}$  result in numerical solution expediency in much the same way as the time derivative term in the artificial compressibility equation. Numerical solution of equations (25) and (26) advances efficiently in time with the use of an ADI solver. Moreover, the  $\epsilon \xi_{\tau}$  and  $\epsilon \eta_{\tau}$  terms afford an opportunity to test the effectiveness of providing a truly dynamic transformation which should move grids in response to the changing physical domain. Evaluation of numerical results is deemed to be the most effective means of determining the appropriate range of values for  $\epsilon$ , hence no a priori estimate for the value of  $\epsilon$  is made.

Having chosen the method for generating the fixed  $\xi$ - $\eta$  computational domain, the differential equations (3), (5), (18), (19), (20), (25) and (26) are transformed to the  $\xi$ - $\eta$  plane where the calculations are to be performed.

Conceptually the following occurs. Flow parameters and mapping functions are evaluated at mesh points in the computational plane by a finite difference solution of the governing equations. The inverse transformation,  $x = x(\xi, \eta, \tau)$  and  $y = y(\xi, \eta, \tau)$ , determined from the image of equations (25) and (26) locates the points in the physical domain which correspond to the computational mesh points at which the flow parameters have been determined.

Transformation of equations (3), (5), (18), (19) and (20) requires that all derivatives be expressed in terms of  $\xi$  and  $\eta$ , the independent variables of the computational domain. The chain rule is applied to obtain

$$(27) \quad f_x = (f_\xi y_\eta - f_\eta y_\xi)/J$$

$$(28) \quad f_y = (f_\eta x_\xi - f_\xi x_\eta)/J$$

$$(29) \quad f_\tau = f(\xi, \eta, \tau) - x_\tau(f_\xi y_\eta - f_\eta y_\xi)/J + y_\tau(f_\xi x_\eta - f_\eta x_\xi)/J$$

where all derivatives on the left hand side of equations (27) to (29) are defined at points in the computational plane. The quantity  $J$ , the Jacobian of the transformation, is given by

$$(30) \quad J = J\left(\frac{x, y, \tau}{\xi, \eta, \tau}\right) = x_\xi y_\eta - x_\eta y_\xi.$$

Expressions for higher order derivatives, obtained by repeated operations, are shown in Appendix C. After considerable manipulation,<sup>1</sup> the relations (27) to (29) transform the differential equations (3), (5), (25) and (26) to the following system:

---

<sup>1</sup> The reader is encouraged to examine the formulae of Appendix C to gain an understanding of the complexity of this calculation.

$$\begin{aligned}
(31) \quad & (J\vec{q})_{\tau} + \{J[\vec{q}(y_{\tau}x_n - x_{\tau}y_n) + \vec{f}y_n - \vec{g}x_n]\}_{\xi} \\
& + \{J[\vec{q}(x_{\tau}y_{\xi} - y_{\tau}x_{\xi}) - \vec{f}y_{\xi} + \vec{g}x_{\xi}]\}_{\eta} \\
& = \frac{1}{\text{Re}} \left[ \frac{y_n}{J} (\overline{D}y_n \vec{q}) + \frac{x_n}{J} (\overline{D}x_n \vec{q}) \right]_{\xi} - \left[ \frac{y_n}{J} (\overline{D}y_{\xi} \vec{q}) \right. \\
& \left. + \frac{x_n}{J} (\overline{D}x_{\xi} \vec{q}) \right]_{\eta} + \frac{1}{\text{Re}} \left[ \frac{x_{\xi}}{J} (\overline{D}x_{\xi} \vec{q}) + \frac{y_{\xi}}{J} (\overline{D}y_{\xi} \vec{q}) \right]_{\eta} \\
& - \left[ \frac{x_{\xi}}{J} (\overline{D}x_n \vec{q}) + \frac{y_{\xi}}{J} (\overline{D}y_n \vec{q}) \right]_{\xi} + E
\end{aligned}$$

$$(32) \quad \varepsilon J^2 x_{\tau} - \alpha x_{\xi\xi} + 2\mu x_{\xi\eta} - \omega x_{\eta\eta} = 0$$

$$(33) \quad \varepsilon J^2 y_{\tau} - \alpha y_{\xi\xi} + 2\mu y_{\xi\eta} - \omega y_{\eta\eta} = 0$$

where  $\vec{q} = [u, v, \beta p]^t$ ,  $\vec{f} = [p + u^2, uv, u]^t$ ,  $\vec{g} = [uv, p + v^2, v]^t$ ,

$$\overline{D} = \begin{bmatrix} 100 \\ 010 \\ 000 \end{bmatrix},$$

$$\alpha = x_n^2 + y_n^2, \quad \mu = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \quad \omega = x_{\xi}^2 + y_{\xi}^2,$$

Several observations should be made. First, all fluid flow boundary conditions are also transformed to the  $\xi$ - $\eta$  plane. Transformation of equations (18) to (20) is detailed in a following section; transformation of the remaining boundary conditions is straight forward. Secondly, equation (31) expresses conservation of fluid mass in the computational plane in a manner analogous to

equation (5) (see Appendix C). Boundary conditions for equations (32) and (33) are defined so that the boundary of  $Q$  maps onto the boundary of  $D$ . This results in the requirements that  $x(0, \eta, \tau) = 0$ ,  $y(0, \eta, \tau) = \eta$ ,  $x(1, \eta, \tau) = 1$ ,  $y(1, \eta, \tau) = \eta$ ,  $x(\xi, 0, \tau) = L\xi$ , and  $y(\xi, 0, \tau) = F(x(\xi, 0, \tau))$  where for simplicity  $Q$  has been defined as the unit square. Although transformation to the  $\xi$ - $\eta$  plane introduces cross derivative terms which complicate the numerical solution, transformation provides the convenience of computing on a uniform rectangular grid and the accuracy of applying boundary conditions along straight lines. Finally, all of the equations (31) to (33), and all transformed boundary conditions are solved by finite difference techniques at each step.

Analytic assessment of the application of numerical mapping equations (32) and (33) is difficult. Indeed, it is standard practice to rely on numerical results to verify that the numerical map yields a reasonable equivalent finite difference grid in the physical plane (i.e., to check the image of the computational grid in the physical plane). Nonetheless, a few observations should be made.

First, the inverse function theorem states that if the Jacobian of the transformation,  $J = x_{\xi} y_{\eta} - x_{\eta} y_{\xi}$ , is singular at some point  $(\xi_0, \eta_0, \tau_0)$ , then the mapping may no longer be one-to-one in some neighborhood of that point. That is, when  $J = 0$  it cannot be assumed that the equivalent finite difference grid in the physical plane is

not composed of looped, crossed, coalesed or similarly degenerate lines. Hence, the numerical scheme should monitor the value of the Jacobian and advise that calculations stop if points are found at which  $J = 0$ .

For calculations in which the Jacobian is everywhere nonzero, the inverse function theorem verifies that the transformation from the computational plane to the physical plane is locally one-to-one. Also, boundary conditions specify that the image of the boundaries of the computational domain define the boundaries of the physical plane. For  $\epsilon > 0$ , the mapping equations (32) and (33) are parabolic, as shown in Appendix E. Thus, equations (32) and (33), for the given boundary conditions, have unique continuous solutions (see Friedman's text [25]), guaranteeing uniqueness of continuous coordinate functions  $x = x(\xi, \eta, \tau)$  and  $y = y(\xi, \eta, \tau)$ . To further corroborate the use of equations (32) and (33), consider the following linearization about the known solution state functions,  $x^\circ$  and  $y^\circ$ . Letting  $x$  and  $y$  denote the numerically determined solutions, set

$$(34) \quad \tilde{x} = x - x^\circ$$

$$(35) \quad \tilde{y} = y - y^\circ$$

and assume that calculations have been successful enough so that  $\tilde{x}$  and  $\tilde{y}$  are small. Make term by term approximations, proceeding, for example as

$$\begin{aligned}
x_n^2 &= (\tilde{x} + x^\circ)_n (\tilde{x} + x^\circ)_n \\
&= (x_n^\circ)^2 + 2\tilde{x}x_n^\circ + O((\tilde{x}_n)^2) \\
&\approx (x_n^\circ)^2 + 2x_n^\circ(x - x^\circ)_n \\
&= 2x_n^\circ x_n - (x_n^\circ)^2
\end{aligned}$$

to obtain the following linear (in terms of the unknowns  $x$  and  $y$ ) parabolic differential equations:

$$\begin{aligned}
(36) \quad & \{\alpha^\circ/[\epsilon(J^\circ)^2]\}x_{\xi\xi} - \{2\mu^\circ/[\epsilon(J^\circ)^2]\}x_{\xi n} \\
& + \{\omega^\circ/[\epsilon(J^\circ)^2]\}x_{nn} \\
& + \{[2x_n^\circ x_{\xi\xi}^\circ - 2x_\xi^\circ x_{\xi n}^\circ + \epsilon y_\xi^\circ x_\tau^\circ]/[\epsilon(J^\circ)^2]\}x_n \\
& + \{[2y_n^\circ x_{\xi\xi}^\circ - 2y_\xi^\circ x_{\xi n}^\circ - \epsilon x_\xi^\circ x_\tau^\circ]/[\epsilon(J^\circ)^2]\}y_n \\
& + \{[2x_\xi^\circ x_{nn}^\circ - 2x_n^\circ x_{\xi n}^\circ - \epsilon y_n^\circ x_\tau^\circ]/[\epsilon(J^\circ)^2]\}x_\xi \\
& + \{[2y_\xi^\circ x_{nn}^\circ - 2y_n^\circ x_{\xi n}^\circ + \epsilon x_n^\circ x_\tau^\circ]/[\epsilon(J^\circ)^2]\}y_\xi - x_\tau = 0
\end{aligned}$$

$$\begin{aligned}
(37) \quad & \{\alpha^\circ/[\epsilon(J^\circ)^2]\}y_{\xi\xi} - \{2\mu^\circ/[\epsilon(J^\circ)^2]\}y_{\xi n} \\
& + \{\omega^\circ/[\epsilon(J^\circ)^2]\}y_{nn} \\
& + \{[2y_n^\circ y_{\xi\xi}^\circ - 2y_\xi^\circ y_{\xi n}^\circ - \epsilon x_\xi^\circ y_\tau^\circ]/[\epsilon(J^\circ)^2]\}x_n \\
& + \{[2x_n^\circ y_{\xi\xi}^\circ - 2x_\xi^\circ y_{\xi n}^\circ + \epsilon y_\xi^\circ y_\tau^\circ]/[\epsilon(J^\circ)^2]\}y_n \\
& + \{[2y_\xi^\circ y_{nn}^\circ - 2y_n^\circ x_{\xi n}^\circ + \epsilon x_n^\circ y_\tau^\circ]/[\epsilon(J^\circ)^2]\}x_\xi \\
& + \{[2x_\xi^\circ y_{nn}^\circ - 2x_n^\circ y_{\xi n}^\circ - \epsilon y_n^\circ y_\tau^\circ]/[\epsilon(J^\circ)^2]\}y_\xi - y_\tau = 0
\end{aligned}$$

where  $\alpha^\circ = (x_n^\circ)^2 + (y_n^\circ)^2$ , etc. (note:  $J^\circ = J = x_\xi y_n - x_n y_\xi \neq 0$ ). The linearized system, equations (36) and (37), satisfies a maximum principle which states that the maximum and minimum values for the functions  $x$  and  $y$  defined by these equations can occur only initially or at points on the boundary of the computational domain. That is, the mapping determined by equations (36) and (37) cannot map interior points of the  $\xi$ - $n$  domain to points of the  $x$ - $y$  domain outside the limits defined by the specified

boundaries. Thus, as long as the Jacobian is nonzero and the linearization of equations (34) and (35) is admissible, it is reasonable to expect that the mapping from the computational plane to the physical plane is locally one-to-one, unique, maps interior points to points within the boundaries and maps boundary points to boundary points. From these arguments, it appears that the numerical transformation should be reasonable.

CHAPTER IV  
NUMERICAL FORMULATION

A computer calculated solution for the transformed equations (31) to (33) requires incorporation of transformed expressions of boundary conditions (18) to (20) in a manner compatible with the numerical scheme. All equations used in the application of the scheme must be expressed as a system of linear algebraic equations. Thus, in addition to giving differential terms an algebraic finite difference expression, the numerical formulation must accommodate other influential characteristics of the governing equations. These equations are nonlinear, multidimensional and coupled. The format of the scheme presented here, similar to those of Briley and McDonald [26-29], deals with these aspects of the problem and includes:

- a. implementation of an additional boundary conditions for the purpose of determining free surface fluid pressure,
- b. linearization of the nonlinear terms by Taylor series expansion about the previous (known) time level,
- c. backward in time, centered in space differencing,
- d. expression of the equations as a system of coupled, linear difference equations,

- e. Douglas-Gunn splitting to generate the block ADI system of one dimensional equations and
- f. partitioning of reducible block submatrices.

The procedure results in two one-dimensional linear difference equations that can be solved efficiently by standard block elimination techniques. All boundary conditions are included in the process described above and the subsequent matrix equations. The scheme requires only one iteration to advance the entire flow one time step. The above formulation allows for direct extension of the numerical method to three dimensional flow problems.

As can be seen by inspecting equations (18) to (20), these governing equations provide no direct way to calculate fluid pressure on the free surface. An additional condition is imposed on the free surface so that free surface fluid pressure may be numerically determined. For the purpose of testing the numerical scheme presented in this paper, the free surface water pressure is equated with the atmospheric pressure, i.e., let

$$(38) \quad p = p_0 \quad \text{on } y = H(x, \tau).$$

Rewriting equations (18) and (19) for reference,

$$(18) \quad (1/Re)\{2u_x H_x - [1 - (H_x)^2](u_y + v_x) - 2H_x v_y\} = 0$$

$$\text{on } y = H(x, \tau)$$

$$(19) \quad (1/Re)\{2u_x (H_x)^2 - 2H_x (u_y + v_x) + 2v_y\}$$

$$= (p - p_0) [1 + (H_x)^2] \quad \text{on } y = H(x, \tau),$$

the assumption of equation (38) incorporates the additional condition that the right hand side of equation (19) is

zero. Equations (18) and (19) may then be simplified by separating the stress into its cartesian components

$$(39) \quad (1/Re)[2u_x H_x - (u_y + v_x)] = 0 \quad \text{on } y = H(x, \tau)$$

$$(40) \quad (1/Re)[(u_y + v_x)H_x - 2v_y] = 0 \quad \text{on } y = H(x, \tau).$$

Transformation of equations (39) and (40) yields the following boundary conditions for the numerical scheme:

$$(41) \quad u_\eta = (1/\omega)(\mu u_\xi - Jv_\xi)$$

$$(42) \quad v_\eta = (1/\omega)(Ju_\xi + \mu v_\xi)$$

$$(43) \quad p = p_0.$$

It should be noted that equations (41) to (43) are defined on a line of constant  $\eta$  which maps to the line  $y - H(x, \tau) = 0$ .

To enhance the incorporation of the free surface boundary condition given by equation (20), a Hirt [30] stability analysis if performed on a simplified version of equation (20). This analysis suggests that numerical instabilities can arise in the finite difference solution of equation (20) because of a diffusion truncation error (see Appendix D for details). Thus, to enhance the computational stability of the free surface calculations, a numerical viscosity term is added. To do this, equation (20) is modified to the form

$$(44) \quad H_\tau + uH_x - v - \gamma H_{xx} = 0 \quad \text{on } y = H(x, \tau)$$

where the quantity  $\gamma$  is defined as the coefficient of artificial viscosity. As suggested in Appendix D,  $\gamma > [\max(u^2)\Delta\tau/2]$  where the maximum is taken over the entire flow domain,  $D_f$ . Finally, transformation of equation (44) results in the boundary condition

$$(45) \quad H_\tau + (u/J)[H_\xi Y_\eta - H_\eta Y_\xi] - v \\ + (\gamma/J)\{[(Y_\eta/J)(Y_\eta H_\xi - Y_\xi H_\eta)]_\xi \\ - [(Y_\xi/J)(Y_\eta H_\xi - Y_\xi H_\eta)]_\eta\} = 0$$

which, like equations (41) to (43) is defined on a line of constant  $\eta$ . This makes it possible to eliminate the free surface height,  $H$ , from equation (45). Simplifying gives

$$(46) \quad Y_\tau + u\left(\frac{y_\xi}{x_\xi}\right) - v + \gamma\left[\frac{x_\xi y_{\xi\xi\xi} - y_\xi x_{\xi\xi\xi}}{x_\xi^3}\right] = 0 .$$

Thus, on the free surface, the present implementation of the numerical scheme solves equations (41) to (43) and (46).

Successful numerical implementation of initial conditions requires computational adaptations. Due to computational difficulties encountered when initializing the calculations with nonzero velocities, an accelerated start is utilized to initialize the fluid flow. Beginning with zero fluid velocity and hydrostatic pressure, an acceleration term is used to increase the velocity terms from zero to their uniform flow values.

Frequently, numerical schemes employ distinct or additional computations to evaluate the free surface boundary values; for example, MAC techniques usually obtain free surface boundary values by interpolation while Shanks [14] determines free surface pressure by application of an iterative procedure. Inclusion of the free surface boundary conditions in the general numerical procedure offers the prospect of increased computational efficiency. Moreover, because the free surface strongly effects the

solution throughout the entire domain, calculation of free surface values simultaneously with the flow field should prove to be advantageous. For this reason, it is anticipated that ultimately this totally implicit method will permit increased time steps and speed solution convergence.

Workers in numerical partial differential equations have successfully applied a variety of techniques to the implicit finite difference solution of nonlinear partial differential equations. Ames [28] illustrates several iterative methods for solving finite difference expressions of nonlinear differential equations. With all of these methods, some efficiency is sacrificed in the iterative calculations; speed and success of convergence to the solution commonly depend on the quality of the initial conditions. The scheme of this paper avoids such difficulties by first linearizing the differential system, and then solving the associated finite difference system.

The method of linearization plays an important role in determining the success an implicit scheme has solving the given differential equations. If the linear approximation gives a poor representation of the nonlinear system then small time steps or repeated iterations may be required to maintain solution accuracy, thus compromising the favorable stability properties of an implicit method. Assuming that the solution is known at the  $n$ th time level, the linearization technique employed here utilizes a Taylor series expansion about the  $n$ th time level to reduce the system to differential equations containing only linear terms in the

unknowns at the  $n+1$ st time step. The procedure requires only that the variables are Taylor series expandable, a restriction present in the use of any finite difference method. Truncation errors arising from the linearization process do not lower the formal order of accuracy of the fully implicit backward in time discretization. The linearization readily extends to nonlinear boundary conditions.

To illustrate the linearization process used in the current numerical scheme, consider the following example with one dependent variable,  $Q$ , and two possibly nonlinear functions of  $Q$ ,  $F$  and  $G$ :

$$(47) \quad Q_\tau = F(Q)G_\xi(Q) .$$

Linearization proceeds in a few steps. First, the time derivative is implicitly discretized:

$$(48) \quad \frac{Q^{n+1} - Q^n}{\Delta\tau} = [F(Q)G_\xi(Q)]^{n+1} + O(\Delta\tau)$$

where  $Q^n = Q(n\Delta\tau)$ , etc. Using chain rule differentiation, the nonlinear term is expanded about the  $n$ th time step:

$$(49) \quad [F(Q)G_\xi(Q)]^{n+1} = \{F(Q)G_\xi(Q)\}^n + \Delta\tau \{F(Q)\frac{\partial}{\partial\xi}[G_Q Q_\tau] + [F_Q Q_\tau]G_\xi(Q)\}^n + O(\Delta\tau)^2 .$$

Again, the time derivatives are discretized to obtain:

$$(50) \quad \frac{Q^{n+1} - Q^n}{\Delta\tau} = [F(Q)G_\xi(Q)]^n + \{[F(Q)]^n \frac{\partial}{\partial\xi}[G(Q)^n(Q^{n+1} - Q^n)] + [F_Q]^n(Q^{n+1} - Q^n)[G_\xi(Q)]^n\} + O(\Delta\tau)$$

where the order of the truncation error is determined by the time differencing in equation (48). Also, taking

$F(Q) = 1$  and  $G(Q) = JQ$  ( $J$  is the Jacobian) gives a linearization of the quantity  $\frac{\partial}{\partial \xi}(JQ)$ . This particular linearization suggests the following treatment for the nonlinear time derivation terms which occur in equation (31):

$$(51) \quad \frac{\partial}{\partial \tau}(JQ)^{n+1} = \frac{(JQ)^{n+1} - (JQ)^n}{\Delta \tau} + 0(\Delta \tau)$$

$$= \frac{J^{n+1}Q^n + J^nQ^{n+1} - 2J^nQ^n}{\Delta \tau} + 0(\Delta \tau).$$

Note that equations (50) and (51) are linear in the unknown,  $Q^{n+1}$  so that application of the above outlined procedure to the system of equations (31) to (33), and (41) to (43), and (46) yields an approximating linear differential system. Applying the described linearization to equations (31) to (34), and (41) to (43) and (46) involves extensive algebraic manipulation and the author found linearization of the vector equation (31) tractable only when each component was analyzed individually.

Following linearization, the numerical formulation discretizes the linear differential system. However, before introducing the finite difference operators, the following notation is defined. Grid points having equal spacings,  $\Delta \xi$  and  $\Delta \eta$  in the  $\xi$  and  $\eta$  directions, respectively, and an arbitrary time step,  $\Delta \tau$ , discretize the computational domain. The subscripts  $i$  and  $j$  and superscript  $n$  index the grid point associated with  $\xi$ ,  $\eta$ , and  $\tau$ , respectively. Thus  $Q_{ij}^n$  denotes  $Q((i-1)\Delta \xi, (j-1)\Delta \eta, n\Delta \tau)$  where  $Q$  represents any one of the dependent variables. To obtain the necessary discrete point representation for the

continuously defined linearized system, use the following finite difference operators:

$$(52) \quad \delta_{\xi} Q^n = \frac{Q_{i+1,j}^n - Q_{i-1,j}^n}{2\Delta\xi} = \frac{\partial Q}{\partial \xi} \Big|_{i,j}^n + O((\Delta\xi)^2)$$

$$(53) \quad \delta_{\xi\xi} Q^n = \frac{Q_{i+1,j}^n - 2Q_{ij}^n + Q_{i-1,j}^n}{(\Delta\xi)^2} \\ = \frac{\partial^2 Q}{\partial \xi^2} \Big|_{ij}^n + O((\Delta\xi)^2)$$

$$(54) \quad \delta_{\xi\eta} Q^n = \frac{Q_{i+1,j+1}^n - Q_{i+1,j-1}^n + Q_{i-1,j-1}^n - Q_{i+1,j-1}^n}{4(\Delta\xi)(\Delta\eta)} \\ = \frac{\partial^2 Q}{\partial \xi \partial \eta} \Big|_{ij}^n + O((\Delta\xi)^2 (\Delta\eta)^2)$$

where  $\Delta\xi = \Delta\eta$  and similar expressions are given for  $\delta_{\eta}$  and  $\delta_{\eta\eta}$ . These difference formulae contain values of the dependent variables at adjacent grid points, hence the finite difference equations define an implicit system of banded matrices. The finite difference operators effect the bandwidth of the system. Operators (52) to (54) which, with the application of an ADI procedure produce (block) tridiagonal matrices, are specified to simplify calculations.<sup>1</sup> Operators with a higher order of accuracy complicate calculations to a degree deemed unwarranted in the current test problem. On the boundaries, one-sided, first order  $O(\Delta\xi, \Delta\eta)$  finite difference operators are applied where necessary to maintain block tridiagonal ADI matrices.

---

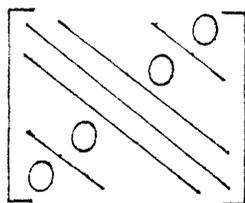
<sup>1</sup> Also, as shown by Roache [2], backward in time (implemented in the linearization) and centered in space differencing results in algebraic expressions which preserve the conservation property for the fluid mass.

It should be noted that linearization and finite differencing procedures may be interchanged; the present order of these operations reduces the proliferation of subscripts.

The equations obtained by the procedure outlined above represent a linearized implicit difference scheme. These equations can be organized according to time and space differences and written in the following form:

$$(55) \quad \bar{A}^n \left( \frac{\bar{Q}^{n+1} - \bar{Q}^n}{\Delta \tau} \right) = \bar{D}_\xi^n \bar{Q}^{n+1} + \bar{D}_\eta^n \bar{Q}^{n+1} + \bar{D}_{\xi\eta}^n \bar{Q}^{n+1} + \bar{B}^n$$

where  $\bar{A}$  is a block (5 x 5 blocks) diagonal matrix;  $\bar{Q}$  is a column vector containing the five unknowns,  $u$ ,  $v$ ,  $p$ ,  $x$ , and  $y$ , at each computational node. The three point difference operators defined in equations (52) to (54) determine  $\bar{D}_\xi$  and  $\bar{D}_\eta$  which are block (5 x 5 blocks) banded matrices. In general, the block submatrices of  $\bar{D}_\xi$  and  $\bar{D}_\eta$  are organized in the following pattern:



where the band width is proportional to the number of computational grid points. Here,  $\bar{D}_{\xi\eta}^n$  contains the finite difference expressions for cross derivative terms.

Implicit treatment of  $\bar{D}_{\xi\eta}^n$ , in which all terms are of order  $1/Re$  or of order  $\epsilon$ , greatly increases computational difficulty. Hence  $\bar{D}_{\xi\eta}^n \bar{Q}^n$  is evaluated in place of  $\bar{D}_{\xi\eta}^n \bar{Q}^{n+1}$ . The term  $\bar{B}$  is a column vector defined by known values from the  $n$ th time level. Each of the matrices  $\bar{A}$ ,  $\bar{D}_\xi$ ,  $\bar{D}_\eta$ , and  $\bar{D}_{\xi\eta}$

has dimension  $5MN \times 5MN$  where  $M$  is the number of nodes in the  $\xi$  direction and  $N$  is the number of nodes in the  $\eta$  direction;  $\bar{Q}$  and  $\bar{B}$  are  $5MN$  vectors. The  $5 \times 5$  submatrices composing  $\bar{A}$ ,  $\bar{D}_\xi$ ,  $\bar{D}_\eta$ , and  $\bar{D}_{\xi\eta}$  and the components of  $\bar{B}$  must be evaluated at each grid point. The present scheme includes all boundary conditions and mapping equations in equation (55).

The bandwidth of the matrices determined by the difference operators in equation (55) presents computational difficulties. Also, if accuracy of the spatial finite differences is increased by reducing the spatial increments,  $\Delta\xi$  or  $\Delta\eta$ , the number of grid points and hence the matrices' bandwidth increases. The system in equation (55) could be solved by several methods. However, the numerical scheme presented here reduces the bandwidth of equation (55) by employing an ADI splitting. The technique used is simply an adaptation to systems of equations of the ADI procedure developed by Douglas and Gunn [32].

For the given two dimensional problem, the ADI scheme comprises two steps, each step treating one spatial derivative implicitly. Specifically, the Douglas-Gunn splitting of equation (54) is given by

$$(56a) \quad \bar{A}^n \left( \frac{\bar{Q}^* - \bar{Q}^n}{\Delta\tau} \right) = \bar{D}_\xi^n \bar{Q}^* + \bar{D}_\eta^n \bar{Q}^n + \bar{B}_1^n$$

$$(56b) \quad \bar{A}^n \left( \frac{\bar{Q}^{n+1} - \bar{Q}^n}{\Delta\tau} \right) = \bar{D}_\xi^n \bar{Q}^* + \bar{D}_\eta^n \bar{Q}^{n+1} + \bar{B}_1^n$$

where  $\bar{B}_1^n = \bar{D}_{\xi\eta}^n \bar{Q}^n + \bar{B}^n$  and  $\bar{Q}^*$  is an intermediate solution. Douglas and Gunn show, under a fairly general assumption,

that the splitting in equation (56) defines consistent approximations to the original difference equation.<sup>2</sup> This consistency makes it appropriate to include the physical boundary conditions at the intermediate step of equation (56) (for reference, see Briley and McDonald [29]). To reduce both the effort involved in the actual programming and the computer storage requirements, equation (56) is rewritten as

$$(57a) \quad [\bar{A}^n - \Delta\tau \bar{D}_\xi^n] (\bar{Q}^* - \bar{Q}^n) = \Delta\tau [\bar{D}_\xi^n \bar{Q}^n + \bar{D}_\eta^n \bar{Q}^n + \bar{B}_1^n]$$

$$(57b) \quad [\bar{A}^n - \Delta\tau \bar{D}_\eta^n] (\bar{Q}^{n=1} - \bar{Q}^n) = \bar{A}^n (\bar{Q}^* - \bar{Q}^n) .$$

The computational algorithm for a single step proceeds as follows:

- a. The first step of the ADI procedure applies equation (57a) on successive rows of grid points in the  $\xi$  direction. With this ordering of nodes, the operator  $[\bar{A}^n - \Delta\tau \bar{D}_\xi^n]$  generates a block tri-diagonal matrix. Since at each node there are five equations, the block system is composed of 5 X 5 submatrices. The solution of (57a) thus

---

<sup>2</sup> The difference equation provides a consistent approximation to the original equation if the difference system agrees with the original equation to within an error that vanishes as  $\Delta\tau \rightarrow 0$ . For this paper, Douglas and Gunn show that, if equation (55) is consistent with equations (31) to (33), (41) to (43) and (46) (no attempt is made to verify consistency here), and if  $(\Delta\tau)^2 \bar{D}_\xi^n \left( \frac{\bar{Q}(\tau + \Delta\tau) - \bar{Q}(\tau)}{\Delta\tau} \right)$  approaches a linear differential operator as  $\Delta\tau \rightarrow 0$ , then the splitting equation (56) is consistent with the original system.

involves Gaussian elimination on the block tridiagonal system and inversion of the  $5 \times 5$  submatrices.

- b. Analogous to step a., the second ADI sweep applies equation (57b) on successive rows of grid points in the  $\eta$  direction.

Discussion of the reduction of the  $5 \times 5$  submatrices is problem dependent and is given in a following section.

## CHAPTER V

### STABILITY ANALYSIS CONSIDERATIONS

When implemented in the solution of time dependent fluid flow equations, implicit time marching schemes, such as the procedure used here, encounter practical limits on the size of the time step. For examples, see Beam and Warming [33]. Calculations generally require some time step restriction to assure that components (in particular any possible errors) of the initial values do not amplify without bound as the numerical solution is advanced through time. For the free surface problem of this paper, stability restrictions may arise from such factors as nonlinearities at interior or boundary points or from the influence of boundary conditions. However, current stability theory cannot directly address these nonlinearities, or in some cases the free surface boundary conditions, thus it is difficult to analytically assess the stability of the scheme proposed here.

An effective tool for examining the stability of a numerical scheme, the Fourier transform method as initially developed by Von Neumann, uses a Fourier series expansion of the unknowns to analyze numerical techniques for solving problems with constant coefficients and spatially periodic boundary conditions (see Richtmyer and Morton [34]). For

the current problem, application of Fourier stability analysis, which evaluates only linear problems, should begin with each step of equation (56), determining the scheme's overall stability from a combination of both steps. The restriction to the linear equation (56), whose time dependent coefficient matrices must be assumed constant, could preclude discovery of stability limitations due to nonlinearities. More importantly, Fourier series expansion of the variables specifies periodicity of the boundary values between the fixed bottom and the free surface. The inconsistency of this assumption with physical reality and the strong influence of the free surface on the solution makes application of Fourier stability analysis to the present problem unreasonable.

Both steps of the ADI scheme in equation (56) or (57) determine matrix equations, each of which has the form  $\bar{M}_1 \bar{Q}^* = \bar{M}_2 \bar{Q}^n$ , where  $\bar{Q}^*$  is the time advanced unknown and  $\bar{Q}^n$  is known. Assuming that for each ADI step the matrices  $\bar{M}_1$  and  $\bar{M}_2$  are constant, two amplification matrices of the form  $\bar{M}_1^{-1} \bar{M}_2$  govern the growth of initial values. The scheme's stability is ultimately determined from the product of these two amplification matrices. Thus, the stability of the scheme depends on the norms of the matrices  $\bar{M}_1$  and  $\bar{M}_2$  (see Mitchell and Griffiths [35]). The problem solved here results in large dimension, 2035 x 2035, asymmetric amplification matrices. Although defining the numerical problem on a coarser finite difference grid would decrease

the size of the amplification matrices which have dimension  $5MN \times 5mn$ , the amplification matrices would still be large. For these large matrices, evaluation of the matrix norm, or even the eigenvalues of the matrix, is a formidable task. Hence, in the present work numerical evaluation of stability is considered more expedient. Preliminary numerical stability results are presented in a following section.

Noting that apart from actual stability analysis, loss of diagonal dominance in the coefficient matrices of equation (56) or (57) increases the possibility of errors arising in the solution, several observations may be made. First,  $\beta$ , the coefficient of artificial compressibility, multiplies the diagonal term of the rows in which  $\beta$  appears. Thus  $\beta$  should not be made arbitrarily small. The parameter  $\epsilon$  has much the same effect on the matrix rows determined by the mapping equations and hence  $\epsilon$  should not be made arbitrarily small. Finally, since the central differences utilized in the scheme presented here do not involve the variable at the central mesh point, the diagonal terms derived from finite difference expression of the free surface stress boundary conditions depend only on the equation linearization and are of order  $\Delta\tau$ ,  $O(\Delta\tau)$ . Hence, the current incorporation of these stress conditions may result in a tacit restriction on  $\Delta\tau$ .

CHAPTER VI  
APPLICATION TO PROBLEM

For the geometry depicted in Figure 1 and the coordinate transformation shown in Figure 2, equations (31) to (33), (41) to (43) govern the numerical problem. To continue the problem formulation requires implementation of the previously outlined numerical procedures. Successful accomplishment of this task demands competence and care on the part of the investigator. Unfortunately, complete discussion of all details pertaining to the successful implementation of the solution scheme to the present problem would burden both the reader and the author. However, examination of several of the linearized equations and associated matrices suffices to indicate the type of manipulations required to finalize the numerical formulation of the governing equations.

As mentioned earlier, linearization of equation (31) proceeds by components (for bookkeeping purposes, these are labeled the  $u$ ,  $v$ , and  $p$  equations, respectively). Linearization of these components and of equation (32) and (33) are shown below in a form applicable to the ADI solver:

$$\begin{aligned}
(58) \quad & \frac{1}{\Delta\tau} \left\{ \left[ J - \frac{2\Delta\tau}{\text{Re}} \left( \frac{x_\xi x_\eta + y_\xi y_\eta}{J} \right) \right]_{\xi\eta} \right\}^n (u^{n+1} - u^n) \\
& + \left\{ (y_\xi u)_\eta - (y_\eta u)_\xi + \frac{\Delta\tau}{\text{Re}} \left( \frac{x_\xi u_\eta + u_\xi x_\eta}{J} \right) \right. \\
& + \left. \frac{(x_\xi y_\eta + x_\eta y_\xi)(u_\xi x_\eta - x_\xi u_\eta) + 2y_\xi y_\eta (u_\xi y_\eta - u_\eta y_\xi)}{J^2} \right\}_{\xi\eta}^n (x^{n+1} - x^n) \\
& + \left\{ (x_\eta u)_\xi - (x_\xi u)_\eta + \frac{\Delta\tau}{\text{Re}} \left( \frac{u_\xi y_\eta + u_\eta y_\xi}{J} \right) \right. \\
& + \left. \frac{(y_\eta x_\xi + y_\xi x_\eta)(u_\eta y_\xi - u_\xi y_\eta) + 2x_\xi x_\eta (u_\eta x_\xi - u_\xi x_\eta)}{J^2} \right\}_{\xi\eta}^n (y^{n+1} - y^n) \} \\
& = \left\{ x_\tau y_\eta - y_\tau x_\eta - 2u y_\eta + x_\eta v - \frac{1}{\text{Re}} \left[ \left( \frac{x_\eta x_\xi + y_\xi y_\eta}{J} \right)_\eta \right] \right\}^n u^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\eta^2 + y_\eta^2}{J} \right] \right\}^n u_\xi^{n+1} + \{ x_\eta u \}^n v^{n+1} - \{ y_\eta \}^n p^{n+1} \\
& + \{ (y_\tau u)_\eta - (uv)_\eta + \frac{1}{\text{Re}} \left[ \frac{2x_\xi u_\eta - x_\eta u_\xi}{J} \right] \right. \\
& + \left. \frac{u_\xi y_\eta (x_\xi x_\eta + y_\eta y_\xi) - u_\eta y_\eta (x_\xi^2 + y_\xi^2)}{J^2} \right\}_\eta^{\eta} x_\eta^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{y_\eta^2 (y_\xi u_\eta - u_\xi y_\eta) + x_\eta y_\eta (u_\eta x_\xi - u_\xi x_\eta)}{J^2} - \frac{u_\eta x_\eta}{J} \right] \right\}_\eta^{\eta} x_\xi^{n+1} \\
& + \{ (p + u^2)_\eta - (x_\tau u)_\eta + \frac{1}{\text{Re}} \left[ \frac{2u_\eta y_\xi - u_\xi y_\eta}{J} \right] \right. \\
& + \left. \frac{u_\eta x_\eta (x_\xi^2 + y_\xi^2) - u_\xi x_\eta (x_\xi x_\eta + y_\eta y_\xi)}{J^2} \right\}_\eta^{\eta} y_\eta^{n+1} \\
& + \frac{1}{\text{Re}} \left[ \frac{x_\eta^2 (u_\xi x_\eta - u_\eta x_\xi) + x_\eta y_\eta (u_\xi y_\eta - u_\eta y_\xi) - u_\eta y_\eta}{J^2} \right]_\eta^{\eta} y_\xi^{n+1} \} \\
& + \left\{ (y_\tau x_\xi - x_\tau y_\xi - x_\xi v + 2u y_\xi - \frac{1}{\text{Re}} \left[ \frac{x_\xi x_\eta + y_\xi y_\eta}{J} \right]_\xi) \right\}_\xi^{\eta} u^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi^2 + y_\xi^2}{J} \right] \right\}_\xi^{\eta} u_\eta^{n+1} - \{ x_\xi u \}^n v^{n+1} + \{ y_\xi \}^n p^{n+1} \\
& + \{ (uv)_\xi - (y_\tau u)_\xi + \frac{1}{\text{Re}} \left[ \frac{2x_\eta u_\xi - u_\eta x_\xi}{J} \right] \right. \\
& + \left. \frac{u_\xi y_\xi (y_\eta^2 + x_\eta^2) - u_\eta y_\xi (y_\xi y_\eta + x_\eta y_\xi)}{J^2} \right\}_\xi^{\eta} x_\eta^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi y_\xi (x_\xi u_\eta - x_\eta u_\xi) + y_\xi^2 (u_\eta y_\xi - u_\xi y_\eta) - \frac{u_\xi x_\xi}{J}}{J^2} \right] \right\}_\xi^{\eta} x_\eta^{n+1}
\end{aligned}$$

$$\begin{aligned}
& + \{ (x_\tau u)_\xi - (p + u^2)_\xi + \frac{1}{\text{Re}} \left[ \frac{2y_n u_\xi - y_\xi u_n}{J} \right. \\
& + \left. \frac{u_n x_\xi (y_\xi y_n + x_n x_\xi) - u_\xi x_\xi (y_n^2 + x_n^2)}{J^2} \right]_\xi \}^n y^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi y_\xi (u_\xi y_n - u_n y_\xi) + x_\xi^2 (u_\xi x_n - x_\xi u_n)}{J^2} - \frac{u_\xi y_\xi}{J} \right]_\xi \right\}^n y_n^{n+1} \Big]_n \\
& - \frac{2}{\text{Re}} \left\{ \frac{y_\xi y_n + x_\xi x_n}{J} \right\}^n u_{\xi n}^{n+1} + \frac{1}{\text{Re}} \left\{ \frac{x_\xi u_n + x_n u_\xi}{J} \right. \\
& + \left. \frac{2y_\xi y_n (u_\xi y_n - u_n y_\xi) + (u_\xi x_n - u_n x_\xi) (y_\xi x_n + x_\xi y_n)}{J^2} \right\}^n x_{\xi n}^{n+1} \\
& + \frac{1}{\text{Re}} \left\{ \frac{y_n u_\xi + y_\xi u_n}{J} \right. \\
& + \left. \frac{2x_\xi x_n (x_\xi u_n - u_\xi x_n) + (u_n y_\xi - u_\xi y_n) (y_\xi x_n + x_\xi y_n)}{J^2} \right\}^n y_{\xi n}^{n+1} \\
& + \{ [2x_\tau u y_\xi - 2y_\tau u x_\xi + 2x_\xi v u - y_\xi p - 2u^2 y_\xi]_n \\
& - [2x_\tau u y_n - 2y_\tau u x_n - 2x_n u v - y_n p - 2u^2 y_n]_\xi \\
& + \frac{1}{\text{Re}} \left[ 2 \left( \frac{x_\xi x_n + y_\xi y_n}{J} \right)_{\xi n} u - 2 \left( \frac{(x_\xi x_n + y_\xi y_n) (u_\xi y_n - u_n y_\xi)}{J^2} \right)_{\xi n} x \right. \\
& \left. - 2 \left( \frac{(x_\xi x_n + y_\xi y_n) (x_\xi u_n - u_\xi x_n)}{J^2} \right)_{\xi n} y \right] \}^n \\
(59) \quad & \frac{1}{\Delta \tau} \left\{ \left[ J - \frac{2\Delta \tau}{\text{Re}} \left( \frac{y_\xi y_n + x_\xi x_n}{J} \right)_{\xi n} \right]^n (v^{n+1} - v^n) \right. \\
& + \left[ (y_\xi v)_n - (y_n v)_\xi + \frac{\Delta \tau}{\text{Re}} \left( \frac{x_\xi v_n + v_\xi x_n}{J} \right. \right. \\
& + \left. \left. \frac{(x_\xi y_n + x_n y_\xi) (v_\xi y_n - v_n y_\xi) + 2x_\xi x_n (v_\xi x_n - v_n x_\xi)}{J^2} \right)_{\xi n} \right]^n (x^{n+1} - x^n) \\
& + \left[ (x_n v)_\xi - (x_\xi v)_n + \frac{\Delta \tau}{\text{Re}} \left( \frac{v_\xi y_n + v_n y_\xi}{J} \right. \right. \\
& + \left. \left. \frac{(y_n x_\xi + y_\xi x_n) (v_n y_\xi - v_\xi y_n) + 2x_\xi x_n (v_n x_\xi - v_\xi x_n)}{J^2} \right)_{\xi n} \right]^n (y^{n+1} - y^n) \\
& = [-\{y_n v\}^n u^{n+1} + \{x_\tau y_n - y_\tau x_n - y_n u + 2x_n v \\
& - \frac{1}{\text{Re}} \left[ \frac{y_\xi y_n + x_\xi x_n}{J} \right]_n \}^n v^{n+1} + \left\{ \frac{1}{\text{Re}} \left[ \frac{y_n^2 + x_n^2}{J} \right] \}^n v_{\xi}^{n+1}
\end{aligned}$$

$$\begin{aligned}
& + \{x_n\}^n p^{n+1} + \{(y_\tau v)_n - (p + v^2)_n + \frac{1}{\text{Re}} \left[ \frac{v_n x_\xi - v_\xi x_n}{J} \right. \\
& + \left. \frac{(x_\xi x_n + y_\xi y_n)(v_\xi y_n - v_n y_\xi)}{J^2} \right]_n\}^n x^{n+1} \\
& + \frac{1}{\text{Re}} \left[ \frac{y_n^2 (y_\xi v_n - y_n v_\xi) + x_n y_n (x_\xi v_n - x_n v_\xi)}{J^2} - \frac{v_n x_n}{J} \right]_n x_\xi^{n+1} \\
& + \{(uv)_n - (x_\tau v)_n + \frac{1}{\text{Re}} \left[ \frac{v_n y_\xi - v_\xi y_n}{J} \right. \\
& + \left. \frac{(v_n x_\xi - v_\xi x_n)(x_\xi x_n + y_\xi y_n)}{J^2} \right]_n\}^n y^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_n^2 (v_\xi x_n - v_n x_\xi) + x_n y_n (v_\xi y_n - y_\xi v_n)}{J^2} \right. \right. \\
& - \left. \left. \frac{v_n y_n}{J} \right]_n y_\xi^{n+1} \right\}_\xi + \{ \{y_\xi v\}^n u^{n+1} + \{y_\xi u + y_\tau x_\xi \\
& - x_\tau y_\xi - 2x_\xi v - \frac{1}{\text{Re}} \left[ \frac{y_\xi y_n + x_\xi x_n}{J} \right]_\xi \}^n v^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi^2 + y_\xi^2}{J} \right] \right\}_n v_n^{n+1} - \{x_\xi\}^n p^{n+1} + \{(p_\tau v^2)_\xi - (y_\tau v)_\xi \\
& + \frac{1}{\text{Re}} \left[ \frac{y_n y_\xi (v_\xi y_n - v_n y_\xi) - x_\xi v_n (J + x_n y_\xi) + x_n v_\xi (2J + x_n y_\xi)}{J^2} \right]_\xi \}^n x^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi y_\xi (x_\xi v_n - x_n v_\xi) + y_\xi^2 (y_\xi v_n - y_n v_\xi)}{J^2} \right. \right. \\
& - \left. \left. \frac{v_\xi x_\xi}{J} \right] \right\}_n x_n^{n+1} + \{(x_\tau v)_\xi - (uv)_\xi \\
& + \frac{1}{\text{Re}} \left[ \frac{v_\xi y_n (2J - y_n x_\xi) + v_n y_\xi (y_n x_\xi - J) + x_\xi x_n (v_n x_\xi - v_\xi x_n)}{J^2} \right]_\xi \}^n y^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{x_\xi^2 (x_n v_\xi - x_\xi v_n) + x_\xi y_\xi (v_\xi y_n - y_\xi v_n)}{J^2} - \frac{v_\xi y_\xi}{J} \right] \right\}_n y_n^{n+1} \Big]_n \\
& - \left\{ \frac{2}{\text{Re}} \left[ \frac{y_\xi y_n + x_\xi x_n}{J} \right] \right\}_n v_{\xi n}^{n+1} \\
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{2(v_\xi y_n - v_n y_\xi)(x_\xi x_n + y_\xi y_n)}{J} \right] \right\}_n x_{\xi n}^{n+1}
\end{aligned}$$

$$\begin{aligned}
& + \left\{ \frac{1}{\text{Re}} \left[ \frac{v_\xi y_\eta + v_\eta y_\xi}{J} \right. \right. \\
& + \left. \frac{2x_\xi x_\eta (v_\eta x_\xi - v_\xi x_\eta) + (v_\eta y_\xi - v_\xi y_\eta)(y_\eta x_\xi + y_\xi x_\eta)}{J^2} \right] \}^n y_{\xi\eta}^{n+1} \\
& + \{ [2x_\tau v y_\xi - 2y_\tau v x_\xi + x_\xi p + 2x_\xi v^2 - 2y_\xi v u]_\eta \\
& + [2y_\tau v x_\eta - 2x_\tau v y_\eta + 2y_\eta u v - x_\eta p - 2x_\eta v^2]_\xi \\
& + \frac{1}{\text{Re}} \left[ 2 \left( \frac{x_\xi x_\eta + y_\xi y_\eta}{J} \right)_{\xi\eta} v - 2 \left( \frac{(x_\xi x_\eta + y_\xi y_\eta)(v_\xi y_\eta - v_\eta y_\xi)}{J^2} \right)_{\xi\eta} x \right. \\
& \left. - 2 \left( \frac{(x_\xi x_\eta + y_\xi y_\eta)(v_\eta x_\xi - v_\xi x_\eta)}{J^2} \right)_{\xi\eta} y \right] \}^n - \frac{1}{\text{Fr}^2}
\end{aligned}$$

$$(60) \quad \frac{1}{\Delta\tau} \{ [\beta J]^n (p^{n+1} - p^n) + [\beta(y_\xi p)_\eta - \beta(y_\eta p)_\xi]^n (x^{n+1} - x^n)$$

$$+ [\beta(x_\eta p)_\xi - \beta(x_\xi p)_\eta]^n (y^{n+1} - y^n) \} = [-\{y_\eta\}^n u^{n+1}$$

$$+ \{x_\eta\}^n v^{n+1} + \{\beta[x_\tau y_\eta - y_\tau x_\eta]\}^n p^{n+1} + \{\beta(y_\tau p)_\eta$$

$$- v_\eta\}^n x^{n+1} + \{u_\eta - \beta(x_\tau p)_\eta\}^n y^{n+1}]_\xi + [\{y_\xi\}^n u^{n+1}$$

$$- \{x_\xi\}^n v^{n+1} + \{\beta[y_\tau x_\xi - x_\tau y_\xi]\}^n p^{n+1}$$

$$+ \{v_\xi - \beta(y_\tau p)_\xi\}^n x^{n+1} + \{\beta(x_\tau p)_\xi - u_\xi\}^n y^{n+1}]_\eta$$

$$- \{ [2\beta(y_\tau p x_\xi - x_\tau p y_\xi) + y_\xi u - x_\xi v]_\eta$$

$$+ [2\beta(x_\tau p y_\eta - y_\tau p x_\eta) + x_\eta v - y_\eta u]_\xi \}^n$$

$$(61) \quad \frac{1}{\Delta\tau} \{ [-\epsilon y_\eta + \Delta\tau \frac{2y_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2}]_{\xi\eta} \}^n (x^{n+1} - x^n)$$

$$+ [\epsilon x_\eta - \Delta\tau \frac{2x_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2}]_{\xi\eta} \}^n (y^{n+1} - y^n) \}$$

$$= \epsilon x_\tau y_\eta^{n+1} - \epsilon y_\tau x_\eta^{n+1} + \left\{ - \frac{y_\eta (y_\eta^2 + x_\eta^2)}{J^2} \right\}^n x_\xi^{n+1}$$

$$+ \left\{ \frac{x_\eta (y_\eta^2 + x_\eta^2)}{J^2} \right\}^n y_\xi^{n+1} - \left\{ \left[ \frac{x_\xi (x_\eta^2 + y_\eta^2)}{J} \right]_\eta \right\}^n y^{n+1}$$

$$\begin{aligned}
& + \left\{ \left[ \frac{y_\xi (x_\eta^2 + y_\eta^2)}{J^2} \right]_\eta \right\}^n x^{\eta+1} \quad \xi + \left\{ \left[ \frac{y_\eta}{J} - \frac{x_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_\xi \right\}^n y^{\eta+1} \\
& + \left\{ \left[ \frac{x_\eta}{J} + \frac{y_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_\xi \right\}^n x^{\eta+1} + \left\{ \frac{x_\eta (x_\xi^2 + y_\xi^2)}{J^2} \right\}^n y^{\eta+1} \\
& - \left\{ \frac{y_\eta (x_\xi^2 + y_\xi^2)}{J^2} \right\}^n x_\eta^{\eta+1} \quad \eta + \left\{ \frac{2y_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right\}^n x_{\xi\eta}^{\eta+1} \\
& - \left\{ \frac{2x_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right\}^n y_{\xi\eta}^{\eta+1} + \{ \epsilon x_\eta y_\tau - \epsilon x_\tau y_\eta + \left[ \frac{x_\eta^2 + y_\eta^2}{J} \right]_\xi \\
& - \left[ \frac{x_\xi x_\eta + y_\xi y_\eta}{J} \right]_\eta + \left[ \frac{2x_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_{\xi\eta} y - \left[ \frac{2y_\eta (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_{\xi\eta} x \}^n
\end{aligned}$$

$$\begin{aligned}
(62) \quad & \frac{1}{\Delta\tau} \left\{ \left[ \epsilon y_\xi - \Delta\tau \left( \frac{2x_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right)_{\xi\eta} \right]^\eta (x^{\eta+1} - x^\eta) \right. \\
& + \left. \left[ -\epsilon x_\xi + \Delta\tau \left( \frac{2y_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J} \right)_{\xi\eta} \right]^\eta (y^{\eta+1} - y^\eta) = \epsilon y_\tau x_\xi^{\eta+1} \right. \\
& - \epsilon x_\tau y_\xi^{\eta+1} + \left\{ \frac{x_\xi (x_\eta^2 + y_\eta^2)}{J^2} \right\}^n y_\xi^{\eta+1} - \left\{ \frac{y_\xi (x_\eta^2 + y_\eta^2)}{J^2} \right\}^n x_\xi^{\eta+1} \\
& - \left\{ \left[ \frac{y_\xi}{J} + \frac{x_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_\eta \right\}^n y^{\eta+1} + \left\{ \left[ -\frac{x_\xi}{J} \right. \right. \\
& + \left. \left. \frac{y_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_\eta \right\}^n x^{\eta+1} \quad \xi + \left\{ \frac{x_\xi (y_\xi^2 + x_\xi^2)}{J^2} \right\}^n y_\eta^{\eta+1} \\
& - \left\{ \frac{y_\xi (y_\xi^2 + x_\xi^2)}{J^2} \right\}^n x_\eta^{\eta+1} - \left\{ \left[ \frac{x_\eta (x_\xi^2 + y_\xi^2)}{J^2} \right]_\xi \right\}^n y^{\eta+1} \\
& + \left\{ \left[ \frac{y_\eta (x_\xi^2 + y_\xi^2)}{J^2} \right]_\xi \right\}^n x^{\eta+1} \quad \eta - \left\{ \frac{2x_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right\}^n y_{\eta\xi}^{\eta+1} \\
& + \left\{ \frac{2y_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right\}^n x_{\eta\xi}^{\eta+1} + \{ \epsilon y_\xi x_\tau - \epsilon x_\xi y_\tau + \left[ \frac{x_\xi x_\eta + y_\xi y_\eta}{J} \right]_\xi \\
& - \left[ \frac{x_\xi^2 + y_\xi^2}{J} \right]_\eta - \left[ \frac{2y_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_{\xi\eta} x + \left[ \frac{2x_\xi (x_\xi x_\eta + y_\xi y_\eta)}{J^2} \right]_{\xi\eta} y \}^n .
\end{aligned}$$

The form of these equations suggests the importance of minimizing computation by establishing a canonical (i.e., simplified) representation for the linearized equations. Equations (58) to (62) incorporate such a simplification. The above equations also demonstrate the need for proficient computer programming.

When necessary, linearization of boundary conditions proceeds much easier as shown by the following linearization for equations (41), (42) and (46):

$$(63) \quad [x_n + y_\xi y_n]^n u_\xi^{n+1} + [x_n y_\xi - y_n]^n v_\xi^{n+1} + [y_n u_\xi + x_n v_\xi - 2u_n y_\xi]^n y_\xi^{n+1} - [1 + (y_\xi)^2]^n u_n^{n+1} + [u_\xi + y_\xi v_\xi]^n x_n^{n+1} + [y_\xi u_\xi - v_\xi]^n y_n^{n+1} + \{v_\xi y_n + 2u_n y_\xi^2 - u_\xi x_n - 2y_\xi u_\xi y_n - 2x_n v_\xi y_\xi\}^n = 0$$

$$(64) \quad [y_n - x_n y_\xi]^n u_\xi^{n+1} + [x_n + y_\xi y_n]^n v_\xi^{n+1} + [y_n v_\xi - x_n u_\xi - 2x_n y_\xi]^n y_\xi^{n+1} - [1 + y_\xi^2]^n v_n^{n+1} + [u_\xi + y_\xi v_\xi]^n y_n^{n+1} + [v_\xi - y_\xi u_\xi]^n x_n^{n+1} + \{2x_n u_\xi y_\xi + 2v_n y_\xi^2 - u_\xi y_n - v_\xi x_n - 2y_\xi v_\xi y_n\}^n = 0$$

$$(65) \quad \frac{1}{\Delta\tau} \{ [x_\xi^3]^n (y^{n+1} - y^n) + [\Delta\tau y_\xi x_\xi^2]^n (u^{n+1} - u^n) - [\Delta\tau x_\xi^3]^n (v^{n+1} - v^n) \} = \{-3x_\xi^2 y_\tau - 2uy_\xi x_\xi + 3vx_\xi^2 - \gamma y_\xi\}^n x_\xi^{n+1} + \{\gamma y_\xi\}^n x_\xi^{n+1} + \{\gamma x_\xi x_\xi - ux_\xi\}^n y_\xi^{n+1} + \{-\gamma x_\xi\}^n y_\xi^{n+1} + \{3x_\xi^3 y_\tau + 2ux_\xi^2 y_\xi - 2vx_\xi^3 + \gamma x_\xi y_\xi\}^n - \gamma y_\xi x_\xi\}^n.$$

As indicated by the above equations, some manipulation may be necessary in adapting the boundary conditions to the form of the matrix equations (56) or (57).

The form of the linearized equations raises a couple of practical questions. The first question concerns the verification of linearization correctness. Symmetries and similarities among terms within the equations provide ample opportunity to certify manipulations by comparison of independently repeated calculations. For example, examine the viscous terms of equations (58) and (59). Reduction of the equations to a canonical form and subsequent examination further corroborates the correctness of the linearized equations.

Currently available numerical solutions of free surface viscous flow problems require some taxing manipulations, performed by the investigator or repeatedly by the computer. Do the inconveniences of linearization make the present scheme unacceptable for general application? Such a question is difficult to address impartially, and, indeed the ultimate applicability of a numerical method should be determined by the ability of the scheme to efficiently provide accurate results for a variety of problems. The solution technique presented here is designed to minimize iterative computations by the computer (cost being the driving factor). This requirement necessitates a certain amount of a priori equation manipulation. Note that linearization of the dynamics equation (31) is by

far the most demanding; when required, boundary condition linearization follows easily. Hence, the scheme's adaptability to a variety of boundary conditions is not compromised by the linearization procedure. Also, the general nature of the present two dimensional transformed equations (31) to (33) makes the linearized fluid dynamics and mapping equations directly applicable to many fluid flow problems, alleviating the necessity of linearization for such applications.

The finite differencing of the linearized equations is accomplished by applying the given finite difference operators within the computer code. Further numerical formulation proceeds to the implementation of the ADI scheme. Recall the large dimensional (5MN x 5MN) of the coefficient matrices in the ADI equation (57). The impracticability of computer storing these large matrices coupled with the simplicity of the Gaussian elimination routine used to solve the block system lead to node-by-node computation. Calculation of coefficients at each computational grid point results in the evaluation of and inversion of combinations of the 5 x 5 submatrices. At interior nodes, the fluid dynamics and mapping equations define the coefficient submatrices for the vector of unknowns,  $[u \ v \ p \ x \ y]^T$  as shown in Table 1.

Examination of the matrices in Table 1 shows that each has the form

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ o & o & o & x & x \\ o & o & o & x & x \end{bmatrix}$$

where  $x$  denotes a nonzero entry. The linear combinations of these matrices appearing in equation (57) also have this form. It is possible to speed computation by partitioning these coefficient matrices and then inverting the  $3 \times 3$  and  $2 \times 2$  blocks. This partition bypasses the computationally more time consuming inversion of the original  $5 \times 5$  blocks.

Table 1

Terms of Submatrices  $\bar{A}' = (A'_{ij})$ ,  $\bar{D}'_{\xi} = (D'_{\xi ij})$ ,  
 $\bar{D}'_{\eta} = (D'_{\eta ij})$ ,  $\bar{D}'_{\xi\eta} = (D'_{\xi\eta ij})$ ,  $\bar{B}' = (B'_i)$

$$A'_{11} = J - 2\Delta\tau\left(\frac{1}{Re}\right)\left[\frac{\alpha}{J}\right]_{\xi\eta}$$

$$A'_{12} = 0$$

$$A'_{13} = 0$$

$$A'_{14} = (y_{\xi} u)_{\eta} - (y_{\xi} u)_{\eta} + \Delta\tau\left(\frac{1}{Re}\right)\left\{\left[\left(\frac{1}{J}\right)(x_{\xi} u_{\eta} + x_{\eta} u_{\xi})\right.\right. \\ \left.\left.+ \left(\frac{1}{J^2}\right)(\mu u_{\xi} y_{\xi} - \omega u_{\eta} y_{\eta} + \alpha u_{\xi} y_{\eta} - \alpha u_{\eta} y_{\xi})\right]_{\xi\eta}\right\}$$

$$A'_{15} = (x_{\eta} u)_{\xi} - (x_{\xi} u)_{\eta} + \Delta\tau\left(\frac{1}{Re}\right)\left\{\left[\left(\frac{1}{J}\right)(u_{\eta} y_{\xi} + u_{\xi} y_{\eta})\right.\right. \\ \left.\left.+ \left(\frac{1}{J^2}\right)(\omega x_{\eta} u_{\eta} - \mu u_{\xi} x_{\xi} + \alpha x_{\xi} u_{\eta} - \alpha u_{\xi} x_{\eta})\right]_{\xi\eta}\right\}$$

$$A'_{21} = 0$$

$$A'_{22} = J - 2\Delta\tau\left(\frac{1}{Re}\right)\left[\frac{\alpha}{J}\right]$$

$$A'_{23} = 0$$

$$A'_{24} = (y_{\xi} v)_{\eta} - (y_{\eta} v)_{\xi} + \Delta\tau\left(\frac{1}{Re}\right)\left\{\left[\left(\frac{1}{J}\right)(x_{\xi} v_{\eta} + v_{\xi} x_{\eta})\right.\right. \\ \left.\left.+ \left(\frac{1}{J^2}\right)(\mu v_{\xi} y_{\xi} - \omega v_{\eta} y_{\eta} + \alpha v_{\xi} y_{\eta} - \alpha v_{\eta} y_{\xi})\right]_{\xi\eta}\right\}$$

$$A'_{25} = (x_{\eta} v)_{\xi} - (x_{\xi} v)_{\eta} + \Delta\tau\left(\frac{1}{Re}\right)\left\{\left[\left(\frac{1}{J}\right)(v_{\xi} y_{\eta} + v_{\eta} y_{\xi})\right.\right. \\ \left.\left.+ \left(\frac{1}{J^2}\right)(\omega v_{\eta} x_{\eta} - \mu v_{\xi} x_{\xi} + \alpha v_{\eta} x_{\xi} - v_{\xi} x_{\eta})\right]_{\xi\eta}\right\}$$

$$A'_{31} = 0$$

$$A'_{32} = 0$$

$$A'_{33} = \beta J$$

$$A'_{34} = \beta[(y_{\xi} p)_{\eta} - (y_{\eta} p)_{\xi}]$$

$$A'_{35} = \beta[(x_{\eta} p)_{\xi} - (x_{\xi} p)_{\eta}]$$

$$A'_{41} = 0$$

$$A'_{42} = 0$$

$$A'_{43} = 0$$

$$A'_{44} = -\epsilon y_{\eta} + 2\Delta\tau\left(\frac{y_{\eta} \alpha}{J^2}\right)_{\xi\eta}$$

Table 1 (Continued)

$$A'_{45} = \epsilon x_{\eta} - 2\Delta\tau \left( \frac{x_{\eta}^{\alpha}}{J^2} \right)_{\xi\eta}$$

$$A'_{51} = 0$$

$$A'_{52} = 0$$

$$A'_{53} = 0$$

$$A'_{54} = \epsilon y_{\xi} - 2\Delta\tau \left( \frac{y_{\xi}^{\alpha}}{J^2} \right)_{\xi\eta}$$

$$A'_{55} = -\epsilon x_{\xi} + 2\Delta\tau \left( \frac{x_{\xi}^{\alpha}}{J^2} \right)_{\xi\eta}$$

$$D'_{\xi 11} = \delta_{\xi} \{ (x_{\tau} y_{\eta} - y_{\tau} x_{\eta} - 2u y_{\eta} + x_{\eta} v) \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\mu}{J} \delta_{\xi} \cdot \right] \right. \\ \left. - \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left( \frac{\alpha}{J} \right)_{\eta} \cdot \right\} \right.$$

$$D'_{\xi 12} = \delta_{\xi} \{ x_{\eta} u \cdot \}$$

$$D'_{\xi 13} = -\delta_{\xi} \{ y_{\eta} \cdot \}$$

$$D'_{\xi 14} = \delta_{\xi} \{ [(y_{\tau} u)_{\eta} - (uv)_{\eta}] \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \left( \frac{\mu}{J^2} \right) (y_{\xi} u_{\eta} - y_{\eta} u_{\xi}) \right] \delta_{\xi} \cdot \right\} \\ + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{1}{J^2} (\mu u_{\xi} y_{\xi} - \alpha u_{\eta} y_{\xi} + J x_{\xi} u_{\eta}) \right]_{\eta} \cdot \right\}$$

$$D'_{\xi 15} = \delta_{\xi} \{ [(p + u^2)_{\eta} - (x_{\tau} u)_{\eta}] \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{1}{J^2} (\alpha u_{\eta} x_{\xi} - \mu u_{\xi} x_{\xi} \right. \right. \\ \left. \left. + J u_{\eta} y_{\xi}) \right]_{\eta} \cdot \right\} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\mu}{J^2} (u_{\xi} x_{\eta} - u_{\eta} x_{\xi}) \right] \delta_{\xi} \cdot \right\}$$

$$D'_{\xi 21} = -\delta_{\xi} \{ y_{\eta} v \cdot \}$$

$$D'_{\xi 22} = \delta_{\xi} \{ (x_{\tau} y_{\eta} - y_{\tau} x_{\eta} - y_{\eta} u + 2x_{\eta} v) \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\mu}{J} \delta_{\xi} \cdot \right] \right. \\ \left. - \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left( \frac{\alpha}{J} \right)_{\eta} \cdot \right\} \right.$$

$$D'_{\xi 23} = \delta_{\xi} \{ x_{\eta} \cdot \}$$

$$D'_{\xi 24} = \delta_{\xi} \{ [(y_{\tau} v)_{\eta} - (p + v^2)_{\eta}] \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\mu}{J^2} (v_{\eta} y_{\xi} \right. \right. \\ \left. \left. - v_{\xi} y_{\eta}) \right] \delta_{\xi} \cdot \right\} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{1}{J^2} (\mu v_{\xi} y_{\xi} - \alpha v_{\eta} y_{\xi} + J v_{\eta} x_{\xi}) \right]_{\eta} \cdot \right\}$$

$$D'_{\xi 25} = \delta_{\xi} \{ [(uv)_{\eta} - (x_{\tau} v)_{\eta}] \cdot \} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\mu}{J^2} (v_{\xi} x_{\eta} - v_{\eta} x_{\xi}) \right] \delta_{\xi} \cdot \right\} \\ + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{1}{J^2} (\alpha v_{\eta} x_{\xi} - \mu v_{\xi} x_{\xi} + J v_{\eta} y_{\xi}) \right]_{\eta} \cdot \right\}$$

$$D'_{\xi 31} = -\delta_{\xi} \{ y_{\eta} \cdot \}$$

$$D'_{\xi 32} = \delta_{\xi} \{ x_{\eta} \cdot \}$$

Table 1 (Continued)

$$D'_{\xi 33} = \beta \delta_{\xi} \{ (x_{\tau} y_n - y_{\tau} x_n) \cdot \}$$

$$D'_{\xi 34} = \beta \delta_{\xi} \{ (y_{\tau} p)_n \cdot \} - \delta_{\xi} \{ v_n \cdot \}$$

$$D'_{\xi 35} = -\beta \delta_{\xi} \{ (x_{\tau} p)_n \cdot \} + \delta_{\xi} \{ u_n \cdot \}$$

$$D'_{\xi 41} = 0$$

$$D'_{\xi 42} = 0$$

$$D'_{\xi 43} = 0$$

$$D'_{\xi 44} = -\delta_{\xi} \left\{ \frac{\mu y_n}{J^2} \delta_{\xi} \cdot \right\} + \delta_{\xi} \left\{ \left( \frac{y_{\xi}^{\mu}}{J^2} \right)_n \cdot \right\}$$

$$D'_{\xi 45} = \delta_{\xi} \left\{ \frac{\mu x_n}{J^2} \delta_{\xi} \cdot \right\} - \delta_{\xi} \left\{ \left( \frac{x_{\xi}^{\mu}}{J^2} \right)_n \cdot \right\}$$

$$D'_{\xi 51} = 0$$

$$D'_{\xi 52} = 0$$

$$D'_{\xi 53} = 0$$

$$D'_{\xi 54} = \epsilon Y_{\tau} \delta_{\xi} \{ \cdot \} + \delta_{\xi} \left\{ \left( \frac{\mu y_{\xi}}{J^2} \right) \delta_{\xi} \cdot \right\} + \delta_{\xi} \left\{ \left[ \frac{x_{\xi}}{J} - \frac{\alpha y_{\xi}}{J^2} \right]_n \cdot \right\}$$

$$D'_{\xi 55} = -\epsilon X_{\tau} \delta_{\xi} \{ \cdot \} - \delta_{\xi} \left\{ \left( \frac{\mu x_{\xi}}{J^2} \right) \delta_{\xi} \cdot \right\} + \delta_{\xi} \left\{ \left[ \frac{y_{\xi}}{J} + \frac{J \alpha x_{\xi}}{J^2} \right]_n \cdot \right\}$$

$$D'_{n 11} = \delta_n \{ [Y_{\tau} x_{\xi} - x_{\tau} y_{\xi} - x_{\xi} v + 2u y_{\xi}] \cdot \} + \left( \frac{1}{Re} \right) \delta_n \left\{ \frac{\omega}{J} \delta_n \cdot \right\} \\ - \left( \frac{1}{Re} \right) \delta_n \left\{ \left( \frac{\alpha}{J} \right)_{\xi} \cdot \right\}$$

$$D'_{n 12} = -\delta_n \{ x_{\xi} u \cdot \}$$

$$D'_{n 13} = \delta_n \{ Y_{\xi} \cdot \}$$

$$D'_{n 14} = \delta_n \{ [(uv)_{\xi} - (y_{\tau} u)_{\xi}] \cdot \} + \frac{1}{Re} \delta_n \left\{ \left[ \frac{1}{J^2} (\alpha u_{\xi} y_n - \omega u_n y_n) \right. \right. \\ \left. \left. + J u_{\xi} x_n \right]_{\xi} \cdot \right\} + \left( \frac{1}{Re} \right) \delta_{\xi} \left\{ \left[ \frac{\omega}{J^2} (u_n y_{\xi} - u_{\xi} y_n) \right] \delta_n \cdot \right\}$$

$$D'_{n 15} = \delta_n \{ [(x_{\tau} u)_{\xi} - (p + u^2)_{\xi}] \cdot \} + \left( \frac{1}{Re} \right) \delta_n \left\{ \left[ \frac{1}{J^2} (\omega u_n x_n - \alpha u_{\xi} x_n) \right. \right. \\ \left. \left. + J y_n u_{\xi} \right]_{\xi} \cdot \right\} + \left( \frac{1}{Re} \right) \delta_n \left\{ \left[ \frac{\omega}{J^2} (u_{\xi} x_n - u_n x_{\xi}) \right] \delta_n \cdot \right\}$$

$$D'_{n 21} = \delta_n \{ Y_{\xi} v \cdot \}$$

$$D'_{n 22} = \delta_n \{ [Y_{\xi} u + y_{\tau} x_{\xi} - x_{\tau} y_{\xi} - 2x_{\xi} v] \cdot \} + \left( \frac{1}{Re} \right) \delta_n \left\{ \frac{\omega}{J} \delta_n \cdot \right\} \\ - \left( \frac{1}{Re} \right) \delta_n \left\{ \left( \frac{\alpha}{J} \right)_{\xi} \cdot \right\}$$

Table 1 (Continued)

$$D'_{n23} = -\delta_n \{x_\xi \cdot\}$$

$$D'_{n24} = \delta_n \{[(p + v^2)_\xi - (y_\tau v)_\xi] \cdot\} + \left(\frac{1}{\text{Re}}\right) \delta_n \left\{ \left[ \frac{1}{J^2} (\alpha v_\xi y_n - \omega v_n y_n + J v_\xi v_n] \cdot \right\} + \left(\frac{1}{\text{Re}}\right) \delta_n \left\{ \left[ \frac{\omega}{J^2} (v_n y_\xi - v_\xi y_n) \right] \delta_n \cdot \right\}$$

$$D'_{n25} = \delta_n \{[(x_\tau v)_\xi - (uv)_\xi] \cdot\} + \left(\frac{1}{\text{Re}}\right) \delta_n \left\{ \left[ \frac{\omega}{J^2} (v_\xi x_n - v_n x_\xi) \right] \delta_n \cdot \right\} + \left(\frac{1}{\text{Re}}\right) \delta_n \left\{ \left[ \frac{1}{J^2} (\omega v_n x_n - \alpha v_\xi x_n + J v_\xi y_n] \cdot \right\}$$

$$D'_{n31} = \delta_n \{y_\xi \cdot\}$$

$$D'_{n32} = -\delta_n \{x_\xi \cdot\}$$

$$D'_{n33} = \beta \delta_n \{[y_\tau x_\xi - x_\tau y_\xi] \cdot\}$$

$$D'_{n34} = -\beta \delta_n \{(y_\tau p)_\xi \cdot\} + \delta_n \{v_\xi \cdot\}$$

$$D'_{n35} = \beta \delta_n \{(x_\tau p)_\xi \cdot\} - \delta_n \{u_\xi \cdot\}$$

$$D'_{n41} = 0$$

$$D'_{n42} = 0$$

$$D'_{n43} = 0$$

$$D'_{n44} = -\epsilon y_\tau \delta_n \{\cdot\} - \delta_n \left\{ \left( \frac{y_n \omega}{x J^2} \right) \delta_n \cdot \right\} + \delta_n \left\{ \left[ \frac{x_n}{J} + \frac{y_n \alpha}{J^2} \right] \cdot \right\}$$

$$D'_{n45} = \epsilon x_\tau \delta_n \{\cdot\} + \delta_n \left\{ \left( \frac{x_n}{J^2} \right) \delta_n \cdot \right\} + \delta_n \left\{ \left[ \frac{y_n}{J} + \frac{x_n}{J^2} \right] \cdot \right\}$$

$$D'_{n51} = 0$$

$$D'_{n52} = 0$$

$$D'_{n53} = 0$$

$$D'_{n54} = \delta_n \left\{ \left( \frac{y_\xi \omega}{J^2} \right) \delta_n \cdot \right\} - \delta_n \left\{ \left[ \frac{y_n \omega}{J^2} \right] \cdot \right\}$$

$$D'_{n55} = -\delta_n \left\{ \left( \frac{x_\xi \omega}{J^2} \right) \delta_n \cdot \right\} + \delta_n \left\{ \left[ \frac{x_n \omega}{J^2} \right] \cdot \right\}$$

$$D'_{\xi n 11} = -2 \left( \frac{1}{\text{Re}} \right) \frac{\alpha}{J} \delta_{\xi n} \{\cdot\}$$

$$D'_{\xi n 12} = 0$$

$$D'_{\xi n 13} = 0$$

$$D'_{\xi n 14} = 2 \left( \frac{1}{\text{Re}} \right) \frac{\alpha}{J^2} (u_\xi y_n - u_n y_\xi) \delta_{\xi n} \{\cdot\}$$

Table 1 (Continued)

$$D'_{\xi\eta 15} = 2\left(\frac{1}{Re}\right) \frac{\alpha}{J^2} (x_{\xi} u_{\eta} - u_{\xi} x_{\eta}) \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 21} = 0$$

$$D'_{\xi\eta 22} = -2\left(\frac{1}{Re}\right) \frac{\alpha}{J} \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 23} = 0$$

$$D'_{\xi\eta 24} = 2\left(\frac{1}{Re}\right) \frac{\alpha}{J^2} (v_{\xi} y_{\eta} - v_{\eta} y_{\xi}) \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 25} = 2\left(\frac{1}{Re}\right) \frac{\alpha}{J^2} (v_{\eta} x_{\xi} - v_{\xi} x_{\eta}) \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 31} = 0$$

$$D'_{\xi\eta 32} = 0$$

$$D'_{\xi\eta 33} = 0$$

$$D'_{\xi\eta 34} = 0$$

$$D'_{\xi\eta 35} = 0$$

$$D'_{\xi\eta 41} = 0$$

$$D'_{\xi\eta 42} = 0$$

$$D'_{\xi\eta 43} = 0$$

$$D'_{\xi\eta 44} = 2 \frac{y_{\eta} \alpha}{J^2} \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 45} = -2 \frac{x_{\eta} \alpha}{J^2} \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 51} = 0$$

$$D'_{\xi\eta 52} = 0$$

$$D'_{\xi\eta 53} = 0$$

$$D'_{\xi\eta 54} = -2 \frac{y_{\xi} \alpha}{J^2} \delta\xi\eta\{\cdot\}$$

$$D'_{\xi\eta 55} = 2 \frac{x_{\xi} \alpha}{J^2} \delta\xi\eta\{\cdot\}$$

$$\begin{aligned} B'_1 = & [2x_{\tau} u y_{\xi} - 2y_{\tau} u x_{\xi} + 2x_{\xi} v u - y_{\xi} p - 2u^2 y_{\xi}]_{\eta} - [2x_{\tau} u y_{\eta} \\ & - 2y_{\tau} u x_{\eta} - 2x_{\eta} u v - y_{\eta} p - 2u^2 y_{\eta}]_{\xi} + \left(\frac{1}{Re}\right) \left(2\left(\frac{\alpha}{J}\right)_{\xi\eta} u \right. \\ & \left. - 2\left[\frac{\alpha}{J^2} (u_{\xi} y_{\eta} - u_{\eta} y_{\xi})\right]_{\xi\eta} x - 2\left[\frac{\alpha}{J^2} (x_{\xi} u_{\eta} - u_{\xi} x_{\eta})\right]_{\xi\eta} y\right) \end{aligned}$$

Table 1 (Continued)

$$\begin{aligned}
B'_2 &= [2x_\tau v y_\xi - 2y_\tau v x_\xi + x_\xi p + 2x_\xi v^2 - 2y_\xi uv]_\eta + [2y_\tau v x_\eta \\
&\quad - 2x_\tau v y_\eta + 2y_\eta uv - x_\eta p - 2x_\eta v^2]_\xi + \left(\frac{1}{Re}\right) \left\{ 2\left(\frac{\alpha}{J}\right)_{\xi\eta} v \right. \\
&\quad \left. - 2\left[\frac{\alpha}{J^2}(v_\xi y_\eta - v_\eta y_\xi)\right]_{\xi\eta} - 2\left[\frac{\alpha}{J^2}(v_\eta x_\xi - v_\xi x_\eta)\right]_{\xi\eta} \right\} - \frac{1}{Fr^2} \\
B'_3 &= -[2\beta y_\tau p x_\xi - 2\beta x_\tau p y_\xi + y_\xi u - x_\xi v]_\eta - [2\beta x_\tau p y_\eta \\
&\quad - 2\beta y_\tau p x_\eta + x_\eta v - y_\eta u]_\xi \\
B'_4 &= \epsilon x_\eta y_\tau - \epsilon x_\tau y_\eta + \left[\frac{\mu}{J}\right]_\xi - \left[\frac{\alpha}{J}\right]_\eta + 2\left[\frac{x_\eta^\alpha}{y_\eta J^2}\right]_{\xi\eta} y - 2\left[\frac{y_\eta^\alpha}{J^2}\right]_{\xi\eta} x \\
B'_5 &= \epsilon y_\xi x_\tau - \epsilon x_\xi y_\tau - \left[\frac{\alpha}{J}\right]_\xi + \left[\frac{w}{J}\right]_\eta + 2\left[\frac{y_\xi^\alpha}{J^2}\right]_{\xi\eta} x - 2\left[\frac{x_\xi^\alpha}{J^2}\right]_{\xi\eta} y
\end{aligned}$$

The terms  $\delta\xi\{\cdot\}$ , etc., indicate that differencing is to be done after matrix multiplication.

CHAPTER VII  
NUMERICAL RESULTS AND DISCUSSION

In the course of developing the numerical method, solutions were calculated for laminar flow in the channel shown in Figure 1 and for an undular bore in a channel with a flat bottom. The primary purpose of these calculations is to demonstrate the feasibility of the overall numerical scheme and to provide initial indications on numerical stability. For this reason, the computations offer only sufficient detail to illustrate the plausibility of the numerical results. Implementation of refinements, such as computational grid spacing control or increased orders of accuracy for boundary conditions, slows orderly development and testing of the numerical technique and so are reserved for future investigations.

For the problem diagramed in Figure 1, calculations were performed on a 37 x 11 grid. The bump on the bottom of the channel had a height corresponding to 20 percent of the total fluid depth. Initially, the nondimensional velocities  $u=0$  and  $v=0$  were prescribed. Over a period of five time steps, an acceleration term was applied in the entire computational domain to increase the velocities to their uniform flow values of  $u=1.0$  and  $v=0.0$ . The complete range of flows tested, characterized by the given initial

conditions and the Froude and Reynolds numbers, gave stable results within the limits of computational resources available to the author. For the Reynolds numbers tested, 3.0  $Re$  100.0, with all other parameters fixed the free surface heights differed by less than 0.2 percent at the end of ten iterations. As expected for viscous flows with the specified free slip fixed boundary condition, changes in the Froude number produced a more noticeable effect on the free surface position than change in the Reynolds number. Figure 3 shows the effect of changes in the Froude number on the free surface height. Case 1 was defined by a Froude number of 0.5; case 2 had  $Fr=1.0$  and in the third case the Froude number was 2.0. For all the flows shown in Figure 3,  $Re = 15.0$ . Unfortunately, the effect of changes in the Froude number shown are reverse from expected trends (see Shanks [14]). For a Froude number of 2.0 the surface movement should be greater than when the Froude number is 0.5 (as the Froude number increases the effect of the gravitational forces decreases and hence the effect on the free surface decreases). Allocated computer time was exhausted before the source of this discrepancy could be located. To indicate the success of the numerical mapping procedure, Figure 4 locates the images of the computational mesh points in the nondimensional physical plane. In Figure 4, the elapsed nondimensional time is 0.32.

At this point only preliminary assessment of the numerical stability of the scheme may be made fairly. In

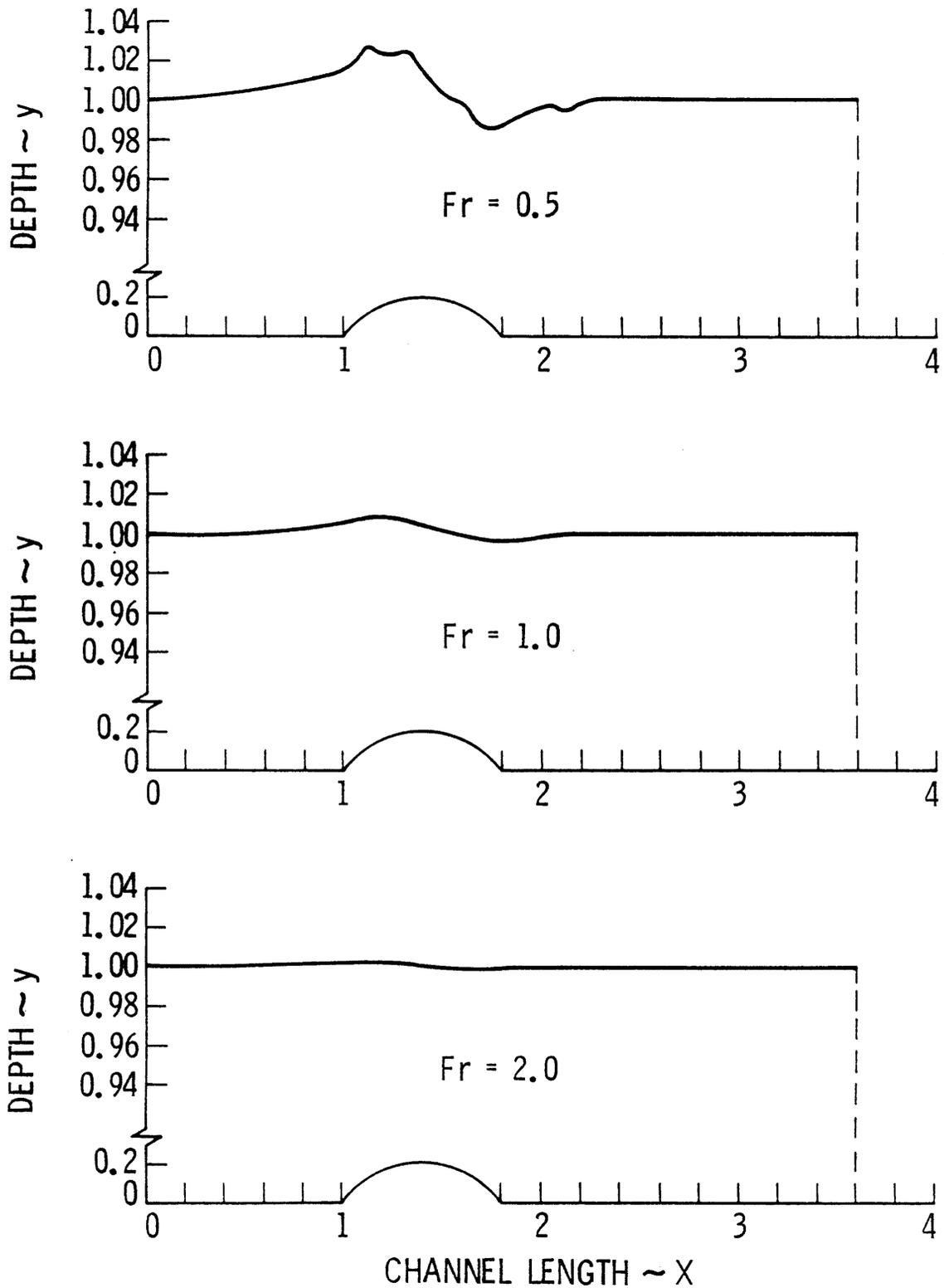


Fig. 3 Free surface elevation for three Froude,  
 $Re = 15.0$ , time = .32 ( $\Delta\tau = .02$ ).

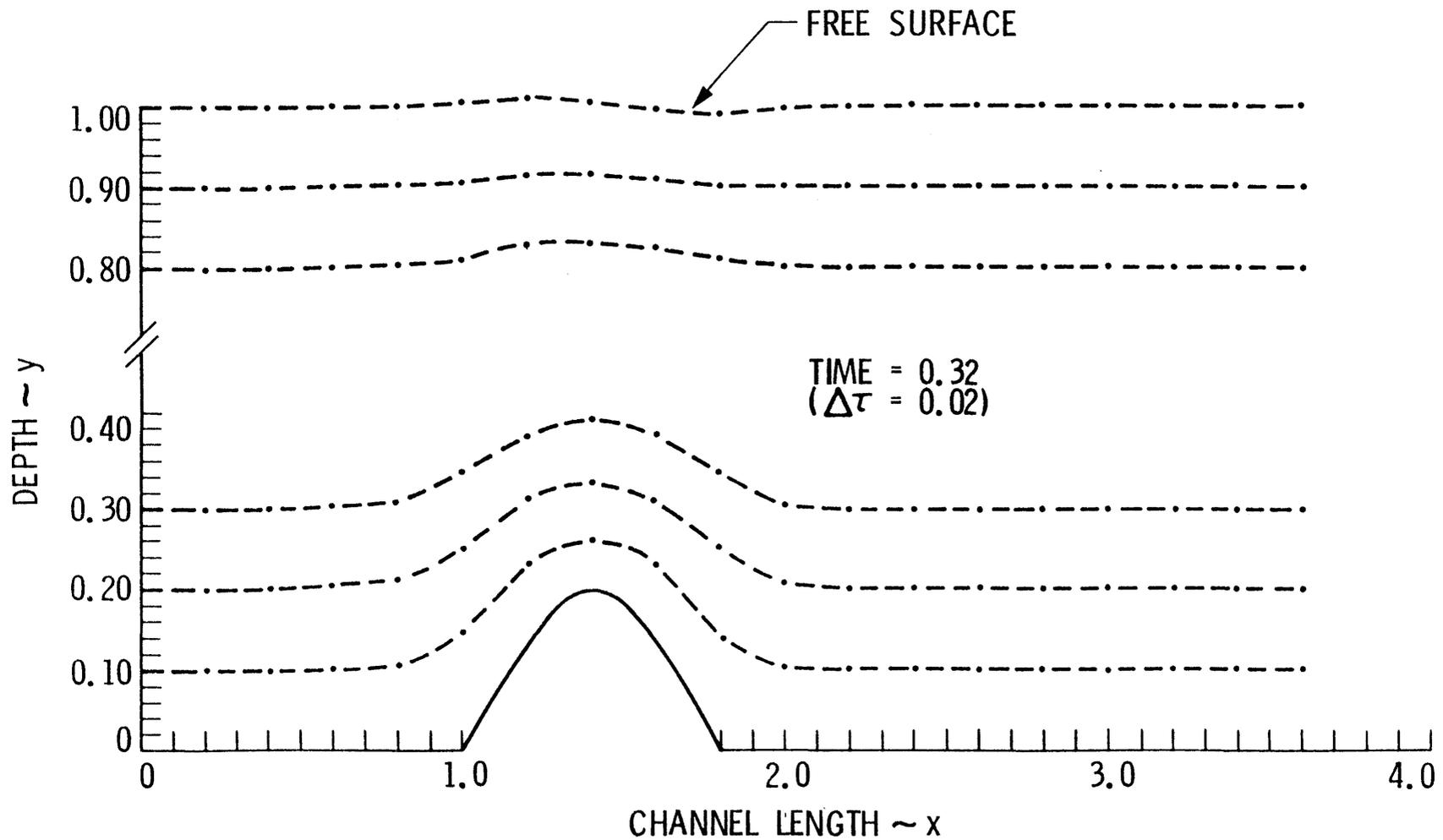


Fig. 4 Numerical grid,  $Fr = 1.0$ ,  $Re = 15.0$ ,  
time = .32 ( $\Delta\tau = .02$ )

the computations performed, the scheme behaved well for time steps up to  $\Delta\tau=0.1$ ; however, for  $\Delta\tau>0.05$ , intermediate use of smaller time steps was necessary to maintain stability. Ultimately, as suggested by Briley and McDonald [23], a sequence of time steps should provide the most efficient solution scheme. Computer resources restricted the author to calculations of sixteen iterations on the 37 x 11 grid. This limitation precluded experimentation with a variety of time steps. For comparison of stability limits, in the explicit free surface MAC scheme described by Welch et al. [36], one stability limit on  $\Delta\tau$  is the Courant-Friedrichs-Lewy (CFL) conditions

$$(66) \quad (g/k)[\tanh(kd)]^{1/2} \Delta\tau < (2\Delta\xi\Delta\eta)/(\Delta\xi + \Delta\eta)$$

where  $k$  is the wave number,  $g$  the force of gravity and  $d$  the fluid depth. A viscosity condition also applies

$$(67) \quad 2\left(\frac{1}{Re}\right)\Delta\tau < (\Delta\xi)^2/[(\Delta\eta)^2 + (\Delta\eta)^2]$$

For the calculations presented in Figure 3, in which  $\Delta\xi=\Delta\eta=0.01$ , the above stability criteria would restrict the maximum allowable time step to approximately 0.0005. Also, Shanks [14], using an implicit free surface scheme, performs most calculations with  $\Delta\tau=0.01$ . Hence, the present scheme appears to provide acceptable numerical stability. Further stability analysis and numerical experimentation are merited.

The values of the two parameters  $\epsilon$  and  $\beta$  appear to have a significant effect on the solution. Although optimum values of these parameters have not been deter-

mined, some observations have been made. Values of  $\epsilon$  on the order of the Reynolds number ( $\epsilon \sim O(Re)$ ) permit the grid to move well with the fluid. Unexpectedly, when  $\epsilon$  was chosen small ( $\epsilon \sim O(Re/100)$ ) the grid disassociated itself from the fluid and numerical instability resulted. Results when  $\epsilon=0$  were also unacceptable. These observations were made for  $Re=15.0$ . It should be noted that  $\epsilon$  multiplies the diagonal term in the matrix rows determined by the mapping equations so that small or zero values of  $\epsilon$  may detrimentally effect the matrix inversion process. Various values of  $\beta$  were also tested to check their effect on the solution. In keeping with the principle of artificial compressibility, it was expected that obtaining physically reasonable solutions would be dependent on small values of  $\beta$ . However, for values of  $\beta$  in the range of  $2 \times 10^{-4}$  to  $2 \times 10^{-2}$ , results were nearly identical. This observation held true as long as the ratio  $\beta/\Delta\tau$  was of the order one ( $\beta/\Delta\tau \sim O(1)$ ). Numerical results suggested that when the ratio  $\beta/\Delta\tau$  was greater than order 10 numerical instability would occur after several iterations. Note that both the time differencing and linearization are of order  $\Delta\tau$  and that  $\beta$  multiplies some diagonal terms in the matrix equations so it should not be surprising that the term  $\beta/\Delta\tau$  may define a stability limit. Because the larger values of  $\beta$  appear to provide better stability for larger values of  $\Delta\tau$ ,  $\beta$  was usually set at the value  $2 \times 10^{-2}$ . In the calculations performed to date, the value of  $\gamma$ ,

free surface coefficient of artificial viscosity, did not have a large effect on the movement of the free surface although results did appear generally more stable when  $\gamma$  was nonzero. The bounds proposed by Hirt seemed reasonable and for  $\Delta\tau=0.02$ ,  $\gamma$  was set to a value of about 0.04. It was noted that when  $\gamma$  was set to values an order of magnitude larger than the minimum suggested by Hirt, the movement of the free surface was severely damped.

Figure 5 shows results at the end of several iterations of a simulation of a bore generated by a moving piston at the left hand end of the channel (flat bottom prescribed). For this problem, the initial velocities are taken to be  $u=0$  and  $v=0$ . Motion of the piston is simulated by accelerating the fluid only at the left hand boundary. In these calculations,  $Re=15.0$  and  $Fr=1.22$ .

The numerical results given above were obtained on a CYBER 170. Each iteration required roughly 1.5 minutes of CPU execution time and approximately 180 kilobytes of central memory. It should be noted that no attempt has been made to optimize computer coding efficiency. Thus, the time per iteration, though currently longer than desired, seems reasonable when compared to iteration times for other methods which solve free surface flow problems. For example, though in the current work computations were

---

<sup>1</sup> Chorin [15] claims that the artificial Mach number,  $M=(Re/\sqrt{\beta})\max((u^2 + v^2)^{1/2})$  must be less than one which suggests that  $\beta$  should be at most on the order of 1/5000 in the present calculations. This restriction did not seem to apply, as Steger and Kutler [17] also appear to have found.

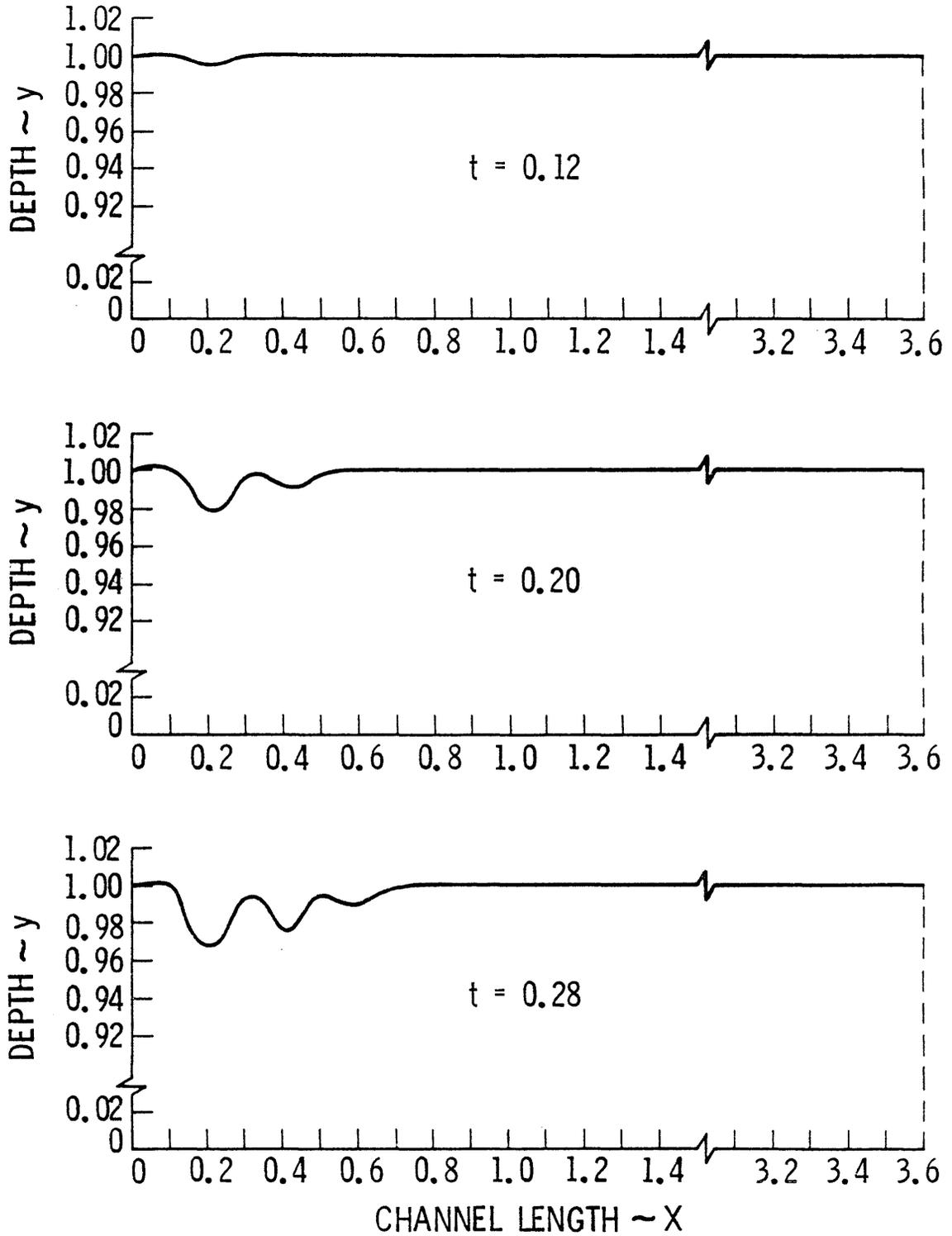


Fig. 5 Time history of movement of the surface of a bore,  $Fr = 1.22$ ,  $Re = 15.0$ ,  $\Delta\tau = .02$ .

not carried to steady state, Shanks [14] required approximately 20 hours (equivalent to 800 iterations at 1.5 minutes/iteration) to calculate free surface problem solutions.

Numerical techniques for solving free surface problems require considerable effort and usually years of effort on the part of several workers to reach a refined level of development. The work presented here attempts only to verify the feasibility of the current scheme. Further numerical testing, improved finite difference accuracy for boundary conditions, grid spacing control or use of sequences of various sized time steps appear to be merited and are suggested for future studies.

## REFERENCES

## REFERENCES

1. Von Kerczek, C. H. (1975). "Numerical Solution of Naval Free-Surface Hydrodynamics Problems," Proceedings of the First International Conference on Numerical Ship Hydrodynamics, David Taylor Naval Research and Development Center, Bethesda, MD.
2. Roache, Patrick J. (1976). Computational Fluid Dynamics, Second Edition, Hermosa Publishers, Albuquerque, N.M.
3. Haussling, H. J., and Van Eseltine, R. T. (1975). "Finite-Difference Methods for Transient Potential Flows with Free Surfaces," Proceedings of the First International Conference on Numerical Ship Hydrodynamics, David Taylor Naval Research and Development Center, Bethesda, MD.
4. Fritts, M. J., Miner, E. W., and Griffen, O. M. (1979). "Numerical Calculations of Wave-Structure Interactions," Computer Methods in Fluids, Pentech Press, London: Plymouth.
5. McCormick, Steve, and Thomas, J. W. (1981). "Multigrid Methods Applied to a Water Wave Problem," Proceedings of the Third International Conference on Numerical Ship Hydrodynamics, Paris, France.
6. Harlow, Francis H., and Welch, J. Eddie. (1965). "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluids with Free Surface," The Physics of Fluids, Vol. 8, No. 12, 2182-2189.
7. Chan, Robert K. C., and Street, Robert L. (1970). "A Computer Study of Finite-Amplitude Water Waves," Journal of Computational Physics, Vol. 6, 68-94.
8. Harlow, Francis H., and Amsden, Anthony A. (1971). "Fluid Dynamics," Report Number LA-4700, Los Alamos Scientific Laboratory, Los Alamos, N.M.

9. Nichols, B. D., and Hirt, C. W. (1973). "Calculating Three-Dimensional Free Surface Flows in the Vicinity of Submerged and Exposed Structures," Journal of Computational Physics, Vol. 12, 234-246.
10. Thames, Frank C. (1975). "Numerical Solution of the Incompressible Navier-Stokes Equations About Arbitrary Two-Dimensional Bodies," Ph.D. Dissertation, Mississippi State University, Mississippi State, MS.
11. Shanks, Samuel P., and Thompson, Joe F. (1977). "Numerical Solution of the Navier-Stokes Equations for 2D Hydrofoils in or Below a Free Surface" Proceedings of the Second International Conference on Numerical Ship Hydrodynamics, University of California, Berkeley, CA.
12. Warsi, Z. U. A., and Thompson, Joe F. (1982). "A Noniterative Method for the Generation of Orthogonal Coordinates in Doubly-Connected Regions," Mathematics of Computation, Vol. 38, No. 158, 501-516.
13. Thompson, Joe F., Thames, Frank C., Mastin, C. W., and Shanks, Samuel P. (1975). "Use of Numerically Generated Body-Fitted Coordinate Systems for Solution of the Navier-Stokes Equations," AIAA 2nd Computational Fluid Dynamics Conference Proceedings, Hartford, CT.
14. Shanks, Samuel P. (1977). "Numerical Simulation of Viscous Flow About Submerged Arbitrary Hydrofoils Using Non-Orthogonal, Curvilinear Coordinates," Ph.D. Dissertation, Mississippi State University, Mississippi State, MS.
15. Aston, Martha B., and Thomas, J. W. (1982). "An Implicit Scheme for Water Wave Problems," Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Use in the Numerical Solution of Partial Differential Equations, NASA Headquarters and Air Force Office of Scientific Research, Nashville, TN.
16. Stokes, J. J. (1957). Water Waves: The Mathematical Theory With Applications. Interscience Publishers, Inc., New York, N.Y., 207-210
17. Chorin, Alexandre Joel. (1967). "A Numerical Method for Solving Incompressible Viscous Flow Problems," Journal of Computational Physics, Vol. 2, 12-26.

18. Chorin, Alexandre Joel. (1968). "Numerical Solution of the Navier-Stokes Equations," Mathematics of Computation, Vol. 22, 745-762.
19. Steger, Joseph L., and Kutler, Paul. (1977). "Implicit Finite-Difference Procedures for the Computation of Vortex Wakes," AIAA Journal, Vol. 15, No. 4, 581-590.
20. Teman, Roger. (1977). Navier-Stokes Equations: Theory and Numerical Analysis, North Holland, New York, N.Y.
21. Shapiro, Ascher, H. (1953). The Dynamics and Thermodynamics of Compressible Fluid Flow, Volume I, The Ronald Press Company, New York, N.Y., 45-69.
22. Steger, J. L. (1977). "Implicit Finite Difference Simulation of Flow About Arbitrary Geometrics with Application to Airfoils," AIAA Paper 77-665, New York, N.Y.
23. Chu, Wen-Hwa. (1971). "Development of a General Finite Difference Approximation for a General Domain, Part I: Machine Transformation," Journal of Computational Physics, Vol. 8, 392-408.
24. Haussling, H. J., and Coleman, R. M. (1981). "A Method for Generation of Orthogonal and Nearly Orthogonal Boundary-Fitted Coordinate Systems," Journal of Computational Physics, Vol. 43, 373-381.
25. Friedman, Avner. (1964). Partial Differential Equations of Parabolic Type, Prentice-Hall, Inc., Englewood Cliffs, N.J., 41.
26. Briley, W. R., and McDonald, H. (1973). "An Implicit Numerical Method for the Multidimensional Compressible Navier-Stokes Equations," Report M911363-6, United Aircraft Corporation, East Hartford, CT.
27. McDonald, H., and Briley, W. R. (1975). "Three-Dimensional Supersonic Flow of a Viscous or Inviscid Gas," Journal of Computational Physics, Vol. 19, No. 2, 150-178.
28. Briley, W. R., and McDonald, H. (1977). "Solution of Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," Journal of Computational Physics, Vol. 24, No. 4, 372-397.

29. Briley, W. R., and McDonald, H. (1980). "On the Structure and Use of Linearized Block Implicit Schemes," Journal of Computational Physics, Vol. 34, No. 1, 54-72.
30. Hirt, C. W. (1968). "Hevristic Stability Theory for Finite-Difference Equations," Journal of Computational Physics, Vol. 2, 339-355.
31. Ames, William F. (1977). Numerical Methods for Partial Differential Equations, Academic Press, New York, N.Y., 82-88.
32. Douglas, Jim, Jr., and Gunn, James. E. (1964). "A General Formulation of Alternating Direction Methods, Part I. Parabolic and Hyperbolic Problems," Numerische Mathematik, Vol. 6, 428-453.
33. Beam, Richard H., and Warming, R. F. (1980). "Alternating Direction Implicit Methods for Parabolic Equations with a Mixed Derivative," SIAM Journal of Scientific and Statistical Computation, Vol. 1, No. 1, 131-159.
34. Richtmyer, R. D., and Morton, K. W. (1967). Difference Methods for Initial-Value Problems, Second Edition, Interscience Publishers, J. Wiley and Sons, New York, N.Y., 60-72.
35. Mitchell, A. R., and Griffiths, D. F. (1980). The Finite Difference Method in Partial Differential Equations, Interscience Publishers, J. Wiley and Sons, New York, N.Y., 40-43.
36. Welch, J. Eddie, Harlow, Francis H., Shannon, John P., and Daly, Bart J. (1965). "The MAC Method: A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surfaces," Report Number LA-3425, Los Alamos Scientific Laboratory, Los Alamos, N.M.

APPENDIXES

## APPENDIX A

To reduce computational complexity, the following approximations are specified on  $y=F(x)$ . Suppose  $y=F(x)$  is constant, then  $u=u_0$  and  $v=0$  from the free slip boundary conditions. From the momentum equation

$$(A1) \quad v_{\tau} + uv_x + vv_y = p_y + (1/Re)(v_{xx} + v_{yy}) - (1/Fr^2) \\ \text{on } y = F(x)$$

it follows that

$$(A2) \quad 0 = -p_y - (1/Fr^2) \quad \text{on } y = F(x).$$

Hence the pressure,  $p$ , may be calculated from

$$(A3) \quad p_y = -(1/Fr^2) \quad \text{on } y = F(x).$$

The reasonable numerical results achieved when this boundary condition is applied are considered sufficient justification for the use of equation (A3) as a first approximation to pressure on  $y = F(x)$ .

APPENDIX B

Transformation of higher order derivations results in the following formulae:

$$(B1) \quad f_{xx} = \frac{y_n^2 f_{\xi\xi} - 2y_\xi y_n f_{\xi n} + y_\xi^2 f_{nn}}{J^2} \\ + \frac{(y_n^2 y_{\xi\xi} - 2y_\xi y_n y_{\xi n} + y_\xi^2 y_{nn})(x_n f_\xi - x_\xi f_n)}{J^3} \\ + \frac{(y_n^2 x_{\xi\xi} - 2y_\xi y_n x_{\xi n} + y_\xi^2 x_{nn})(y_\xi f_n - y_n f_\xi)}{J^3}$$

$$(B2) \quad f_{yy} = \frac{(x_n^2 f_{\xi\xi} - 2x_\xi x_n f_{\xi n} + x_\xi^2 f_{nn})}{J^2} \\ + \frac{(x_n^2 y_{\xi\xi} - 2x_\xi x_n y_{\xi n} + x_\xi^2 y_{nn})(x_n f_\xi - x_\xi f_n)}{J^3} \\ + \frac{(x_n^2 x_{\xi\xi} - 2x_\xi x_n x_{\xi n} + x_\xi^2 x_{nn})(y_\xi f_n - y_n f_\xi)}{J^3}$$

$$(B3) \quad f_{xy} = \frac{[(x_\xi y_n + x_n y_\xi) f_{\xi n} - x_\xi y_\xi f_{nn} - x_n y_n f_{\xi\xi}]}{J^2} \\ + f_\xi \left[ \frac{(x_\xi y_{nn} - x_n y_{\xi n})}{J^2} + \frac{(x_n y_n^J - x_\xi y_n^J)}{J^3} \right] \\ + f_n \left[ \frac{(x_n y_{\xi\xi} - x_\xi y_{\xi n})}{J^2} + \frac{(x_\xi y_n^J - x_n y_\xi^J)}{J^3} \right]$$

## APPENDIX C

Before demonstrating that equation (31) implies conservation of mass analogous to equation (5), note the following. Since the unit normal to the graph of  $f(x,y)=c$  ( $c$  a constant) is given by

$$(C1) \quad \vec{n}(f) = \frac{\nabla f}{|\nabla f|}$$

then, applying equations (27) and (28), the unit normal to a line of constant  $\xi$  is

$$(C2) \quad \vec{n}(\xi) = \frac{y_\eta \vec{i} + x_\eta \vec{j}}{\sqrt{\alpha}}$$

and the unit normal to a line of constant  $\eta$  is

$$(C3) \quad \vec{n}(\eta) = \frac{-y_\xi \vec{i} + x_\xi \vec{j}}{\sqrt{\omega}}$$

Now, consider the third component of equation (34):

$$(C4) \quad (J\beta p)_\tau + \{J[\beta p(y_\tau x_\eta - x_\tau y_\eta) + uy_\eta - vx_\eta]\}_\xi \\ + \{J[\beta p(x_\tau y_\xi - y_\tau x_\xi) + uy_\xi - vx_\xi]\}_\eta = 0.$$

Letting

$$(C5) \quad \vec{F}_1 = \begin{bmatrix} u - \beta p x_\tau \\ v - \beta p y_\tau \end{bmatrix}$$

equation (D4) reduces to

$$(C6) \quad (J\beta p)_\tau + \text{div}(J\vec{F}) = 0$$

where

$$\vec{F} = \begin{bmatrix} \vec{F}_1 \cdot (\sqrt{\alpha} \vec{n}(\xi)) \\ \vec{F}_1 \cdot (\sqrt{\omega} \vec{n}(\eta)) \end{bmatrix}.$$

Equation (C6) expresses the following:

Within a volume,  $W$  ( $W=\Delta x \Delta y$ ), the rate of change of the quantity  $J\beta p$  (a scalar multiple of  $\beta p$ ) equals the rate at

which a scalar factor of the vector  $\vec{F}$  is crossing the boundary of  $W$ .

Such an equation is called a conservation law and is analogous to equation (5). Indeed, examination of the components of  $\vec{F}$  shows that  $\vec{F}$  is composed of components of  $\vec{F}_1$  normal to lines of constant  $\xi$  and normal to lines of constant  $\eta$ . The vector  $\vec{F}_1$  contains the usual flux terms  $u$  and  $v$ . Note that  $\vec{F}_1$  also contains terms which account for any changes in  $\rho p$  due to changes in the volume  $W$ . These terms include the quantities  $x_\tau$  and  $y_\tau$ .

## APPENDIX D

Hirt stability analysis reduces a finite difference equation to a differential equation by expansion of the difference equation in a Taylor series. The lowest order terms from the series must represent the original differential equation. Higher order terms, called truncation errors, are examined for their effect on the stability of the differencing scheme.

To study the stability of an implicit differencing of equation (20), consider the simplified equation

$$(D1) \quad h_{\tau} = -uh_x + v$$

where the velocities  $u$  and  $v$  are taken to be constant.

Suppose a numerical solution for equation (D1) is computed from an implicit finite difference scheme utilizing backward-in-time and centered-in-space differencing. The equation determining the numerical solution for equation (D1) is

$$(D2) \quad \frac{h_i^{n+1} - h_i^n}{\Delta\tau} = -u \frac{h_{i+1}^{n+1} - h_{i+1}^n}{2\Delta x} + v$$

where  $\Delta\tau$  is the time increment,  $\Delta x$  is the spatial increment,  $h_i^n = h(i\Delta x, n\Delta\tau)$ , etc. Assuming that each term in equation (D2) is a continuous function of  $x$  and  $\tau$ , expand the terms in a Taylor series about the point  $(i\Delta x, (n+1)\Delta\tau)$  to obtain

$$(D3) \quad h_{\tau} + \frac{\Delta\tau}{2} h_{\tau\tau} + O(\Delta\tau^2) = -uh_x + v + O(\Delta x^2).$$

Hirt stability analysis assumes the behavior of the solution of equation (D2) (i.e., the numerical solution of equation (D1)) is similar to the behavior of the solution of equation (D3), which is examined analytically. Note that as  $\Delta x \rightarrow 0$  and  $\Delta \tau \rightarrow 0$  in equation (D3), the approximation of equation (D3) to equation (D1), improves which indicates that the solution is correct in this limiting case. However, for  $\Delta x > 0$  and  $\Delta \tau > 0$ , retaining the first order terms in the Taylor series expansion yields

$$(D4) \quad h_{\tau} = -uh_x + v - \frac{u^2 \Delta \tau}{2} h_{xx}.^1$$

Examine equation (D4), derived from equation (D2), to assess the stability of the differencing scheme applied in equation (D2). Apply the relation in equation (D1) to rewrite equation (D4)

$$(D5) \quad h_{\tau} = -uh_x + v - \frac{u^2 \Delta \tau}{2} h_{xx}.$$

Note that comparison of equations (D1) and (D5) shows that equation (D5) has an additional truncation diffusion term,  $\frac{u^2 \Delta \tau}{2} h_{xx}$ . The inclusion of this diffusion term in equation (D5) indicates that the solution of equation (D2) tacitly contains a diffusion term. Setting  $c = \frac{u^2 \Delta \tau}{2}$ , the solution of equation (D5), with appropriate boundary conditions (w.l.o.g.  $h(\Delta \tau) = h(L, \tau) = 0$ ) contains terms of the form

---

<sup>1</sup> This approximation is suggested by expansion of the Fourier components of  $h = h(x, \tau)$  in terms of  $\Delta x$  and  $\Delta \tau$ . See Hirt [22] for a complete discussion. A less formal justification for retaining only first order terms comes from the observation that terms including higher order derivatives are generally smaller than the lower order terms.

$$\exp\left[\left(\frac{cn^2\pi^2}{L^2} + \frac{u^2}{4c}\right)\tau\right] \exp\left[-\frac{u}{2c}x\right] \sin\left(\frac{n\pi}{L}x\right)$$

which grow exponentially in time, for  $c = \frac{u^2\Delta\tau}{2} > 0$ . To

assure a nongrowing stable solution of equation (D5) it is necessary to subtract from  $c$  a term, call it  $\gamma$ , such that  $c - \gamma < 0$ . Thus, for stability, incorporate in both equation (D5) and equation (D2) a diffusion term with coefficient

$$(D6) \quad \gamma > \frac{u^2\Delta\tau}{2} .$$

The term  $\gamma$  is called the coefficient of artificial viscosity. Inclusion of an artificial viscosity term in equation (D2) precludes the inclusion of the same term in equation (D1) which becomes

$$(D7) \quad h_\tau = -uh_x + v + h_{xx}$$

where  $\gamma$  is evaluated from relation (D6). The form of equation (D7) determines the form of equation (44) within the body of the text.

APPENDIX E

The general second order quasi-linear partial differential equation of the form

$$(E1) \quad Lu = \sum_{i,j=1}^2 a_{ij}(x,y,\tau,u) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^2 b_i(x,y,\tau,u) \frac{\partial u}{\partial x_i} - \frac{\partial u}{\partial \tau} = 0$$

is parabolic in domain D if for  $(x,y,\tau) \in D$  the matrix

$$(E2) \quad \bar{M} = \begin{bmatrix} a_{11}(x,y,\tau,u) & a_{12}(x,y,\tau,u) \\ a_{21}(x,y,\tau,u) & a_{22}(x,y,\tau,u) \end{bmatrix}$$

is positive definite. We have the following.

LEMMA E: If, in the domain D, the operator (E1) is parabolic with  $a_{12} = a_{21}$ , then the operator does not change type under the general coordinate transformation

$$(E3) \quad \begin{bmatrix} \xi \\ \eta \\ \tau \end{bmatrix} = \begin{bmatrix} (x,y,\tau) \\ (\eta,y,\tau) \\ \tau \end{bmatrix}$$

at all points for which the Jacobian,  $J = \xi_x \eta_y - \xi_y \eta_x$ , is nonzero. PROOF: First note that if  $J \neq 0$  then, from equations (27) and (28)  $J = x_\xi y_\eta - x_\eta y_\xi \neq 0$ . Also, since the matrix  $\bar{M}$  is positive definite,  $a_{11} > 0$  and

$$a_{11} a_{22} - (a_{12})^2 > 0$$

$$a_{22} > \frac{(a_{12})^2}{a_{11}} \geq 0$$

$$(E4) \quad \left(\frac{a_{22}}{a_{11}}\right) > \left(\frac{a_{12}}{a_{11}}\right)^2 \geq 0.$$

Note that inequality (E4) holds at all points of D.

Applying equations (27) and (28) and the relations of Appendix B, transformation of Lu to  $\xi$ - $\eta$  coordinates results in

$$\begin{aligned} \bar{L}u &= \left\{ \frac{a_{11}y_{\eta}^2 - 2a_{12}x_{\eta}y_{\eta} + a_{22}x_{\eta}^2}{J^2} \frac{\partial^2 u}{\partial \xi^2} \right. \\ &\quad \left. + 2 \left[ \frac{a_{12}(x_{\xi}y_{\eta} + x_{\eta}y_{\xi}) - a_{11}y_{\xi}y_{\eta} - a_{22}x_{\xi}y_{\eta}}{J^2} \right] \frac{\partial^2 u}{\partial \xi \partial \eta} \right. \\ (E5) \quad &\left. + \frac{a_{11}y_{\xi}^2 - 2a_{12}x_{\xi}y_{\xi} + a_{22}x_{\xi}^2}{J^2} \frac{\partial^2 u}{\partial \eta^2} + \sum_{i=1}^2 \bar{b}_i(\xi, \eta, \tau, u) \frac{\partial u}{\partial \xi_i} - u_{\tau} = 0 \right. \end{aligned}$$

where the coefficients  $\bar{b}_i$  may be calculated at the reader's leisure. To verify that  $\bar{L}u$  is parabolic, it is sufficient to show that the matrix

$$(E6) \quad \bar{M} = \frac{1}{J^2} \begin{bmatrix} a_{11}y_{\eta}^2 - 2a_{12}x_{\eta}y_{\eta} & a_{12}(x_{\xi}y_{\eta} + x_{\eta}y_{\xi}) \\ +a_{22}x_{\eta}^2 & -a_{11}y_{\xi}y_{\eta} - a_{22}x_{\xi}x_{\eta} \\ a_{12}(x_{\xi}y_{\eta} + x_{\eta}y_{\xi}) & a_{11}y_{\xi}^2 - 2a_{12}x_{\xi}y_{\xi} \\ -a_{11}y_{\xi}y_{\eta} - a_{22}x_{\xi}x_{\eta} & +a_{22}x_{\xi}^2 \end{bmatrix}$$

is positive definite. Consider

$$\begin{aligned} &\frac{1}{J^2} (a_{11}y_{\eta}^2 - 2a_{12}x_{\eta}y_{\eta} + a_{22}x_{\eta}^2) \\ &= \frac{1}{J^2} \left[ y_{\eta}^2 - 2\left(\frac{a_{12}}{a_{11}}\right) x_{\eta}y_{\eta} + \left(\frac{a_{22}}{a_{11}}\right) x_{\eta}^2 \right] \\ &> \frac{1}{J^2} \left[ y_{\eta}^2 - 2\left(\frac{a_{12}}{a_{11}}\right) x_{\eta}y_{\eta} + \left(\frac{a_{12}}{a_{11}}\right)^2 x_{\eta}^2 \right] \\ &= \left[ \frac{1}{J} \left( y_{\eta} - \frac{a_{12}}{a_{11}} x_{\eta} \right) \right]^2 \geq 0. \end{aligned}$$

Also

$$\begin{aligned} \det (\bar{M}) &= \frac{1}{J^2} \{ [a_{11}y_{\eta}^2 - 2a_{12}x_{\eta}y_{\eta} + a_{22}x_{\eta}^2][a_{11}y_{\xi}^2 - 2a_{12}x_{\xi}y_{\xi} + a_{22}x_{\xi}^2] \\ &\quad - [a_{12}(x_{\xi}y_{\eta} + x_{\eta}y_{\xi}) - (a_{11}y_{\xi}y_{\eta} + a_{22}x_{\xi}x_{\eta})]^2 \} \\ &= \frac{1}{J^2} \{ [a_{11}y_{\eta}^2 + a_{22}x_{\eta}^2][a_{11}y_{\xi}^2 + a_{22}x_{\xi}^2] - [a_{11}y_{\xi}y_{\eta} + a_{22}x_{\xi}x_{\eta}]^2 \\ &\quad - a_{12}^2 J^2 \} = a_{11}a_{22} - a_{12}^2 > 0 \end{aligned}$$

These relations hold at all points in the image of  $D$  so that  $\bar{M}$  is positive definite in the image of  $D$ . It follows that  $\bar{L}u$  is parabolic. Q.E.D. To show that equations (32) and (33) are parabolic, note that equations

$$(25) \quad \epsilon_{\xi\tau} - \nabla_{\xi}^2 = 0$$

$$(26) \quad \epsilon_{\eta\tau} - \nabla_{\eta}^2 = 0$$

(with  $\epsilon > 0$ ) are parabolic. Note that LEMMA E could also be applied to the operator of equation (3) to verify that the transform of equation (3) is also parabolic. It is encouraging to know that the general character of these second order equations is not changed under the coordinate system transformation.