A method for simulation of viscous, non-linear, free-surface flows *

Ben R. Hodges Robert L. Street Yan Zang

Environmental Fluid Mechanics Laboratory, Department of Civil Engineering,
Stanford University, Stanford, CA 94305-4020

ABSTRACT

Presented is a numerical method for simulating free-surface flows through solution of the time-dependent, incompressible, Navier-Stokes equations and the non-linear dynamic and kinematic boundary conditions. The numerical method uses boundary-fitted curvilinear coordinates with a finite-volume, time-splitting, approximate-factorization method formulated in primitive variables on a non-staggered grid. The pressure Poisson equation is solved using a multigrid technique. A new extension of domain decomposition methods is developed which involves splitting the free surface from the fluid volume for the iterative enforcement of the pressure equation and the dynamic boundary condition. A derivation in curvilinear coordinates of the Eulerian kinematic boundary condition is presented and is used for advancing the free surface in a Crank-Nicolson formulation. Development of the numerical method is presented for three dimensions; preliminary results are given from two-dimensional non-linear simulations of standing waves.

NOMENCLATURE

This paper uses tensor notation with the Einstein summation convention implied unless otherwise specifically noted.

a	wave amplitude
B_i	discrete operator for pressure gradient
C_i	discrete operator for convective terms
d	depth
D_α	: $\alpha = 1,2,3$; discrete operators for approximate-factorized diffusive terms
D_I	implicit discrete operator for diagonal diffusive terms
D_E	explicit discrete operator for off-diagonal diffusive terms
e_{ij}	rate of strain tensor
F_{ij}	flux tensor in Cartesian momentum equation
F_i^q	flux tensor in curvilinear momentum equation
F	function representing curvilinear space position of free surface
g	acceleration due to gravity
g^{qr}	contravariant metric tensor
g_{qr}	covariant metric tensor
H	free surface height measured in physical space
$\mathcal{H}$	free surface height measured in curvilinear space
I	identity matrix
J	Jacobian
k	wave number
L	wave length
L_1, L_2	discrete linear operators for dynamic boundary condition
n_i	normal unit vector
p	reduced dynamic pressure
P	pressure
Q_i	discrete operator for grid motion
S_i	source of discretized momentum equation
S_1, S_2	discrete source terms for dynamic boundary condition
t	time (at fixed location in physical space)
t_i	tangent unit vector

* *20th Sym. on Naval Hydrodynamics,* Nat'l. Acad. Press, Washington, D.C. pp. 791-809, 1996.

T wave period
u_i Cartesian velocity
u_i^* intermediate Cartesian velocity
U^q contravariant velocity
U^{*q} intermediate contravariant velocity
$\mathcal{U}$ characteristic velocity of wave
x_i Cartesian (physical space) coordinates
$\dot{X}^q$ grid volume flux
Z free surface vertical coordinate in physical space

Greek Symbols

δ_{ij} Kronecker delta
Δt time step
Δs thickness of free surface
γ gradient of pressure variable (ϕ) at free surface
μ dynamic viscosity
ν kinematic viscosity
ϕ pressure variable
ρ density
ψ second derivative of pressure variable (ϕ) at free surface
τ time (at fixed location in computational space)
ξ^q curvilinear coordinates
ω wave frequency

Subscripts

i, j, k indices for Cartesian or covariant vectors
$S+$ quantity computed in air at free surface
$S-$ quantity computed in water at free surface
$S-1$ quantity computed at first cell center inside free surface
$S-2$ quantity computed at second cell center inside free surface

Superscripts

q, r, s, t indices for contravariant vectors
n, $n+1$ discrete time step

1 INTRODUCTION

Simulations of flows with free-surface effects have been of interest to the numerical community both for their real-world engineering applications and for the challenges which moving-boundary problems present. The physical phenomena that have been modelled using free-surface numerical techniques cover a wide range of areas, including water waves, viscous surface films, bubble dynamics, vortex/free-surface interactions, and geophysical flows. The numerical methods used have been as varied as the phenomena modelled, and include boundary integral methods, spectral methods, finite element methods and finite volume methods. To date, no single approach has been proven superior, and the complexities of free surface phenomena almost guarantee that there will never be one perfect method for all free-surface problems.

In this paper, we develop a technique that is suitable for addressing free-surface wave problems; where the primary characteristic of the flow is nonlinear viscous wave motions of a contiguous free surface. Numerical simulations of this type of free-surface flow face several fundamental problems, including the need for accurate free-surface advancement, enforcement of the non-linear dynamic boundary condition, and practical computation in three dimensions. Both finite element and finite volume methods have been used in the past to study free-surface wave motions [1]. We shall, however, limit our introductory discussion to finite volume methods, which appear to have more promise for future simulation of fully-turbulent flows.

The primary purpose of this paper is to present a numerical method for free-surface flow simulation that incorporates a new method for enforcing the non-linear dynamic boundary condition and uses a curvilinear coordinate derivation of the kinematic boundary condition to advance the free surface.

The new approach to the dynamic boundary condition is to computationally separate the free surface and the fluid volume, then link them through an iterative scheme. This strategy is borrowed from the method of domain decomposition, where solutions on adjacent do-

mains are iterated until the intergrid conditions converge. In our case, rather than enforcing a complicated dynamic boundary condition on the upper boundary of the fluid domain, a simple Dirichlet boundary condition is enforced and refined iteratively with separate computations of the dynamic boundary condition. This "surface decomposition" method may have its ultimate use in coupled simulations of air-water interfaces with decomposed domains consisting of the air volume, the free surface, and the water volume. In such an application, the free surface would be the communication channel for iteratively enforcing the boundary conditions on the two fluids.

The derivation of the Eulerian kinematic boundary condition in this paper is based upon a Taylor-series expansion in curvilinear space, and is similar to the derivation of the Eulerian kinematic boundary condition in physical space by Mei [2]. The curvilinear derivation is advantageous because it allows the simulation of free surfaces which do not remain single-valued in physical space.

Previous finite volume simulations of free-surface flows can roughly be grouped into two categories: 1) "fixed grid" simulations where the governing equations are discretized in physical space on a fixed Cartesian grid while the free-surface moves within the grid; and 2) "moving grid" simulations where a boundary-conforming grid that moves with the free surface is generated.

Most fixed-grid simulations trace their ancestry to the marker-and-cell (MAC) method of Harlow and Welch [3], which uses moving marker particles to track the position of the free surface on a fixed, Cartesian grid. Modified versions of this method are still useful [4] [5] [6]; however, the fundamental drawback of MAC simulations is that the boundary of the computational domain does not lie on the boundary of the fluid. The partially empty cells along the boundary make it difficult to conserve mass and accurately invoke the dynamic boundary condition.

An interesting and relatively new approach using a fixed grid is the "level set" interface technique [7] [8], which is suitable for simulating a coupled domain of two fluids with an immiscible interface. This technique does not explicitly track the position of the free-surface interface, but instead defines a smooth function for the distance from each fixed grid point to the free surface. The simulation solves for this level-set function along with the fluid flow equations. The position of the interface is computed by interpolation from level-set values at the fixed grid points. The level-set technique has an interesting ability to handle flows where the topology of the interface may be changing, such as with combining and dividing bubbles. The major drawback of this method appears to be in the handling of the dynamic boundary condition. As illustrated by Sussman *et al.* [7], the dynamic boundary condition cannot be directly discretized on the free surface in a level-set method because the surface is not given a discrete representation. Instead, the dynamic boundary condition is represented by a smoothed delta function which depends on prescribing a "thickness" of the interface that is greater than the spatial discretization. This requires the discretization be extremely fine, or an unrealistically thick interface be prescribed. We suspect this method may not prove practical for simulating the complicated dynamic boundary condition of a free-surface wave that includes transport of scalars on the surface and spatial variations in the surface-tension coefficient. However, an intriguing possibility worthy of future investigation is a method that combines a level-set solution in computational space (to capture bubble effects) with a boundary-fitted moving grid (to compute a discretized, free-surface wave). This combination might allow efficient computation of breaking waves with bubble effects and surfactant transports on the free surface.

Moving-grid simulations are generally designed with the boundary of the computational domain coincident with the physical domain. This provides a framework for enforcing the dynamic boundary condition directly on the boundary of the computational domain. Most moving-grid simulations generate a structured system of boundary-fitted curvilinear coordinates to map the Cartesian coordinates of points in physical space to a regular orthogonal grid in "computational space". This approach is discussed in greater detail in §2.1 of this paper. An unusual exception to this approach has been developed by Hino *et al.* [9] for steady-state simulations using an unstructured, moving grid that is discretized wholly in physical space.

Two different methods of curvilinear-coordinate grid generation have been used in finite-difference free-surface simulations. One approach [10] [11] [12] is to simplify the computation of grid motion by using what might be termed a "restricted" boundary-fitted grid. In this scheme, the free surface is typically repre-

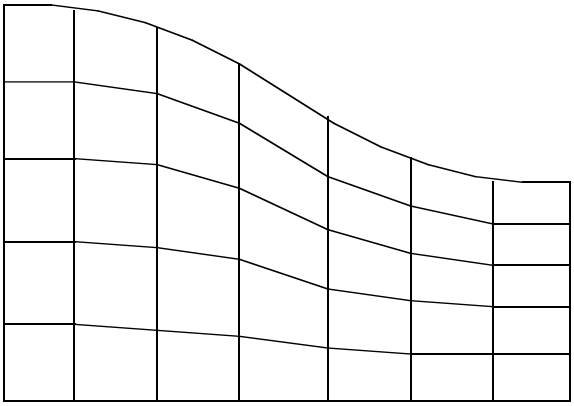

Figure 1: Restricted curvilinear coordinate grid

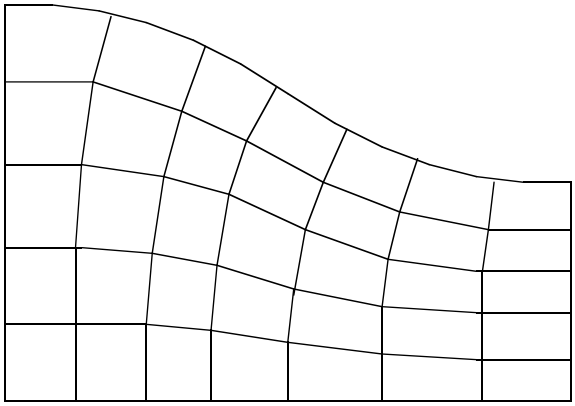

Figure 2: Generalized curvilinear coordinate grid

sented by a series of particles whose horizontal Cartesian coordinates (x, y) are fixed and whose vertical Cartesian coordinate (z) follows the free surface motion along fixed vertical grid lines, as shown in figure (1).

A second approach to moving-grid generation is to use "generalized" boundary-fitted curvilinear coordinates [13] [14]. This method generates smoothly curved grid lines in physical space and allows the grid to be tailored to the requirements of the fluid domain, as shown in figure (2). A generalized curvilinear grid is preferable to a restricted grid because of the ability of the former to handle large surface deformations with minimum grid skewness. This is discussed in greater depth in §2.2.

A number of different numerical techniques have been used to advance the free surface in the solution of viscous free-surface problems. Fully-implicit techniques have been developed for restricted curvilinear coordinates [10], for generalized curvilinear coordinates [14], and for unstructured grids [9] [15]. These techniques simultaneously solve a coupled set of equations consisting of the Navier-Stokes equations, the kinematic and dynamic boundary conditions, and one or more grid generation equations. The advantage of the fully-implicit technique is that the free-surface advance is coupled directly to the flow solution and the dynamic boundary condition, which is arguably more accurate than uncoupled methods. The disadvantage is that significant computational complexity is involved in the coupling of grid generation to the flow solution. To date, the only three-dimensional fully-implicit approach for the Navier-Stokes equations found in the literature is limited to steady-flow solutions [15].

A three-dimensional approach for unsteady flow which appears to effectively advance the free surface without a fully-implicit coupling is that of Dommermuth [11]. His method uses a third-order Runge-Kutta time integration of a kinematic boundary condition based on Helmholtz decomposition.

To avoid the problems associated with simultaneous solution of the flow and grid, many free-surface simulations use explicit free-surface advancement [6] [13] [12] [16] [17]. Explicit techniques generally suffer from the accumulation of error in the free-surface advance over long simulation times; however, the explicit Eulerian technique is a stable and effective method for conducting shorter simulations to investigate phenomena and develop numerical techniques.

Three different forms of the Eulerian kinematic boundary condition are found in the literature: 1) numerical enforcement of the physical space Eulerian kinematic boundary condition [9] [10] [11] [15]; 2) computation of a curvilinear transformation of the physical space Eulerian kinematic boundary condition [6] [13] [12] [14]; and 3) rotation of the Cartesian space frame to enforce the physical space Eulerian kinematic boundary condition in a more suitable orientation [17]. The first two methods are unusable for waves which do not remain single-valued in physical space. The third method, rotation of the coordinate system, does allow computation of a free-surface that is not single-valued in physical space; however, the surface must remain single-valued in the rotated Cartesian space frame, which eliminates the method from use with overturning waves. Most of the approaches to the kinematic boundary condition track the surface by the movement of particles that are required to

move vertically in physical space; this has some drawbacks in the accuracy of the free surface advance for steep waves (see §2.5).

There has been a wide range of numerical flow solvers used in free-surface simulations. As of this date, no method can be said to be superior and we shall not try to debate the pros and cons of the various approaches in this paper. However, in order to put our method in perspective, it is worthwhile to briefly mention some of the approaches that other authors have taken. Dommermuth [11] has implemented a three-dimensional technique that uses fourth- and sixth-order finite differences with a third-order Runge-Kutta scheme. This method is unique in that it uses Helmholtz decomposition to split the governing equations into irrotational and vortical flow parts, which provides the free-surface advance through an equation involving the solenoidal velocity potential rather than the Cartesian or curvilinear velocity. Park *et al.* [6], Hinatsu [12], and Miyata *et al.* [13] use time-splitting methods that are discretized on a staggered grid. Hino *et al.* [9] have developed an artificial-compressibility multigrid method that is suitable for steady-state solutions about ship hull forms. Ohring and Lugt's [14] fully-implicit method uses artificial compressibility to solve for unsteady flow in two dimensions. Wang and Leighton [16] have developed a spectral method that discretizes the Navier-Stokes equations in a vertical-velocity, vertical-vorticity formulation that is suitable for periodic boundary condition problems. An entirely different approach was taken by Zhan and Zhaoshun [18], who investigated the characteristics of a drift current with regular waves by decomposing the surface motion into phase-averaged and fluctuating parts.

The numerical approach in this paper is similar to those of Park *et al.* [6], Hinatsu [12], and Miyata *et al.* [13] in the use of finite-volume discretization of primitive variables with a time-splitting technique. Our numerical approach differs by: 1) advancing the free surface in a method that is not limited to single-valued waves; 2) applying a non-staggered grid to reduce storage requirements for metric terms; and, 3) utilizing a multigrid solver for solution of the pressure Poisson equation with iterative enforcement of the dynamic boundary condition. Our numerical method is an adaptation of the method developed by Zang [19] for fixed-boundary problems which has been shown to be second-order accurate in space and time as well as computationally efficient in three dimensions. The kinematic boundary condition enforcement is discretized by the "time-splitting implicit" method of Chan and Street [20].

This is an interim report on a project whose goal is the development of a three-dimensional, time-dependent, Navier-Stokes simulation of finite-amplitude progressive water waves over an imposed current with non-linear dynamic and kinematic boundary conditions. Presented in this paper are the three-dimensional mathematical and numerical foundation of the method and preliminary two-dimensional simulation results for standing waves in a rectangular basin.

The main body of this paper is divided into four sections, beginning with the mathematical formulation and discretization of the governing equations, §2, followed by a description of the numerical method, §3, a description of the surface decomposition method, §4, and a summary of results from numerical simulations, §5.

2 MATHEMATICAL FORMULATION AND DISCRETIZATION

2.1 Computational Domain and Coordinate Mapping

For moving-grid numerical simulations, "boundary-fitted curvilinear coordinates" are often used. This technique is not unique to moving-grid problems, but has been applied to fixed-grid, finite-difference methods with complicated boundary shapes. The boundary-fitted curvilinear coordinate method is based on the concepts of tensor analysis and coordinate transformation that can be found in textbooks such as Aris [21]. Numerical application of boundary-fitted curvilinear coordinates involves developing a coordinate system which matches the boundaries in physical space and maps to a regular orthogonal grid in computational space.

The distortions of the curvilinear coordinate system as viewed from physical space, figure (2), are measured as metric terms. The metric terms are created when the governing equations are transformed into a regular orthogonal grid in computational space. Using this technique, an irregular domain in physical space can be discretized with a structured (e.g. finite difference) numerical method on a regular grid in

computational space while maintaining the simulation boundary coincident with the physical boundary. The main disadvantages of this technique are that the governing equations are more complicated when transformed to computational space and the requirement to store metric terms can overwhelm computer memory. However, improvements in computer speed and memory storage have made the use of curvilinear coordinate transformations more practical, and they can be considered a standard technique for simulations with non-moving grids and irregular boundaries.

The extension of boundary-fitted curvilinear coordinate methods from fixed-boundary to moving-boundary problems is relatively straightforward and is described by Thompson *et al.* [22]. Numerical implementation, however, can be computationally costly because a new grid and new metric terms must be computed at each time step. Implementation of boundary-fitted curvilinear coordinates requires that transformation operators be used to transform the governing equations from physical space to computational space. The operators are derived from the chain rule for partial differential equations. For a moving grid these operators can be presented as [22]:

$$\frac{\partial}{\partial x_j} = \frac{\partial \xi^q}{\partial x_j}\frac{\partial}{\partial \xi^q} \tag{1}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\partial x_j}{\partial \tau}\frac{\partial \xi^q}{\partial x_j}\frac{\partial}{\partial \xi^q} \tag{2}$$

where x_j, with $j = 1, 2, 3$ are the physical space coordinates; ξ^q with $q = 1, 2, 3$ are the computational space coordinates; $\partial/\partial t$ is a time derivative taken at a fixed point in physical space; and $\partial/\partial\tau$ is a time derivative taken at a fixed point in computational space, and repeated subscript/superscript combinations imply summation.

2.2 Grid generation

Boundary-fitted curvilinear coordinate grids can be generated either using restricted or generalized methods as discussed in the introduction to this paper. Restricted boundary-fitted curvilinear coordinates are a useful simplification as long as the free surface deformations remain small. For steep waves, restricted coordinates provide a highly skewed grid: the grid parallelpipeds will be significantly distorted from the ideal rectangular shape. Such grid skewness is undesirable since it generally results in a decrease in accuracy [22]. The use of generalized boundary-fitted coordinates can avoid this problem by using a grid generation method that minimizes grid skewness.

Restricted curvilinear coordinate systems also suffer from a lack of boundary orthogonality. While it is not possible to generate a completely orthogonal grid for an arbitrarily-shaped, three-dimensional domain, it is possible to generate a grid that is orthogonal to the boundaries. There are distinct advantages in the implementation of numerical techniques with boundary orthogonality. As shown by Zang [23] for a fixed grid, a non-orthogonal boundary in a finite-volume method requires a pressure boundary condition as well as a velocity boundary condition for the Poisson pressure equation to prevent an inconsistent solution. The use of boundary orthogonality removes the requirement for the pressure boundary condition. Furthermore, in deriving a discrete form of the dynamic boundary condition, boundary orthogonality provides significant simplification by removing two of the three skew metric terms at the free-surface. That is, for a free surface $\xi^3 = 1$, boundary-orthogonality provides $g^{13} = g^{23} = 0$.

Generalized boundary-fitted curvilinear coordinates can be produced through algebraic methods (i.e. interpolation from the boundaries) or the solution of a set of partial differential equations (typically a Laplace or Poisson equation) [22]. Algebraic, generalized, grid-generation methods are an improvement over the restricted-coordinate technique since the grid developed can be boundary orthogonal and grid skewness can be reduced. Algebraic methods have low computational cost since they use direct solution rather than iteration techniques. However, algebraic methods do not necessarily produce the optimum grid with minimum skewness, thus they might be considered inferior to partial differential equation methods for generating curvilinear coordinates. A Poisson-solution method will generate a grid with minimum skewness in generalized coordinates, but has the disadvantage of requiring iterative smoothing to obtain a final grid. Depending on the shape of the domain, a Poisson grid-generation method can become the major driver of computational time for a free-surface simulation.

In order to have control over grid skewness, we have incorporated into our code sections from the EAGLE grid generation package developed at Mississippi State University [24]. This code uses an algebraic generation method

(transfinite interpolation) as an input to a Poisson smoother, while leaving the user control over the level of smoothing and the maximum number of iterations (which may be set to zero for purely algebraic computation of generalized curvilinear coordinates). The free-surface is represented by a cubic spline over which the ends of grid lines may freely move so as to obtain a smooth grid with boundary orthogonality.

2.3 Non-staggered grid

The governing equations are discretized using the non-staggered grid technique developed by Zang [19]. This method solves for the pressure and the three Cartesian components of velocity at the grid centers and the normal volume flux through grid face. The main advantage of this technique over a staggered-grid method is the savings of storage for the metric terms. Zang's non-staggered grid method requires the storage of one set of metrics, consisting of nine vector surface area components $\left(J^{-1}\partial\xi^q/\partial x_i\right)$, six non-trivial mesh skewness components $\left(J^{-1}g^{qr}\right)$, and the inverse Jacobian, or cell volume $\left(J^{-1}\right)$ for each control volume. In contrast, a staggered-grid method may require up to seven sets of metric terms for each control volume [25] [26].

To adapt the non-staggered grid method for use in free-surface simulations, a full set of metric terms is defined on each cell face of the free surface. This allows the dynamic boundary condition to be discretized with greater accuracy without significantly impacting the overall data storage requirements.

2.4 Navier-Stokes equations

The time-dependent, constant-density, incompressible Navier-Stokes equations in physical space can be written as:

$$\frac{\partial u_i}{\partial t} + \frac{\partial F_{ij}}{\partial x_j} = 0 \tag{3}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{4}$$

where $u_i : i = 1,3$ are the Cartesian (physical space) components of velocity, and the Cartesian momentum flux tensor (F_{ij}) is:

$$F_{ij} = u_j u_i + p\delta_{ij} - \nu\frac{\partial u_i}{\partial x_j} \tag{5}$$

where ν is the kinematic viscosity, δ_{ij} is the Kronecker delta, and the reduced dynamic pressure (p) is related to the density (ρ), total pressure (P) and the vertical Cartesian coordinate (x_3) by:

$$P = \rho\left(p - gx_3\right) \tag{6}$$

In order to simulate a flow with a free surface in boundary-fitted curvilinear coordinates, equations (1) and (2) are used to transform the physical space Navier-Stokes equations into computational space. To complete the transformation, we must utilize the metric identity [22]:

$$\frac{\partial}{\partial\xi^q}\left(J^{-1}\frac{\partial\xi^q}{\partial x_i}\right) \equiv 0 \tag{7}$$

and the conservation of space [27],

$$\frac{\partial}{\partial\tau}\left(J^{-1}\right) - \frac{\partial}{\partial\xi^q}\left(J^{-1}\frac{\partial\xi^q}{\partial x_j}\frac{\partial x_j}{\partial\tau}\right) = 0 \tag{8}$$

By applying equations (1), (2), (7) and (8) to equations (3) and (4) it is possible to formulate the Navier-Stokes equations in time-dependent boundary-fitted curvilinear coordinates as,

$$\frac{\partial}{\partial\tau}\left(J^{-1}u_i\right) + \frac{\partial}{\partial\xi^q}\left(J^{-1}F_i^q\right) = 0 \tag{9}$$

$$\frac{\partial}{\partial\xi^q}\left(J^{-1}U^q\right) = 0 \tag{10}$$

where the curvilinear momentum flux tensor (F_i^q) is:

$$F_i^q = \left[U^q - \dot{X}^q\right]u_i + \frac{\partial\xi^q}{\partial x_i}p - \nu g^{qr}\frac{\partial u_i}{\partial\xi^r} \tag{11}$$

and other curvilinear quantities are defined as:

$$J^{-1} = det\left|\frac{\partial x_i}{\partial\xi^s}\right| \tag{12}$$

$$U^q = \frac{\partial\xi^q}{\partial x_j}u_j \tag{13}$$

$$\dot{X}^q = \frac{\partial\xi^q}{\partial x_j}\frac{\partial x_j}{\partial\tau} \tag{14}$$

$$g^{qr} = \frac{\partial\xi^q}{\partial x_j}\frac{\partial\xi^r}{\partial x_j} \tag{15}$$

Following the method developed by Zang [19], to discretize the momentum equation we apply the explicit 2nd-order Adams-Bashforth method to the convective terms and the off-diagonal viscous terms, with the implicit Crank-Nicolson scheme for the diagonal viscous terms. The addition of the free surface to Zang's method

requires a grid flux term which accounts for the convective motion of the grid. An explicit Euler discretization in the velocity is used for the grid flux term with an average of the time (n) and $(n+1)$ metrics. The pressure is removed from the momentum equation by a predictor-corrector method and a new pressure variable (ϕ) is defined. A further simplification is made by using a second-order accurate approximate factorization on the left hand side of the discretized momentum equation. The resulting system can be presented as:

1. predictor step

$$\left(I - D_1^{n+1}\right)\left(I - D_2^{n+1}\right)\left(I - D_3^{n+1}\right)\left(u_i^* - u_i^n\right) = S_i \tag{16}$$

2. pressure Poisson equation

$$\frac{\delta}{\delta\xi^q}\left(J^{-1}g^{qr}\frac{\delta\phi}{\delta\xi^r}\right)^{n+1} = \frac{1}{\Delta t}\frac{\delta}{\delta\xi^q}\left(J^{-1}U^{*q}\right) \tag{17}$$

3. corrector step

   (a) for the Cartesian velocity (on cell centers):

$$u_i^{n+1} = u_i^* + \left[\frac{\Delta t}{J^{-1}}B_i\left(\phi\right)\right]^{n+1} \tag{18}$$

   (b) for the normal component of contravariant velocity (on cell faces):

$$\left(U^q\right)^{n+1} = U^{*q} - \Delta t\left(g^{qr}\frac{\delta\phi}{\delta\xi^r}\right)^{n+1} \tag{19}$$

where the pressure variable (ϕ) is related to the reduced pressure (p) by:

$$B_i\left(p\right) = \left[J^{-1} - \frac{\Delta t}{2}D_I\right]\frac{B_i\left(\phi\right)}{J^{-1}} \tag{20}$$

the source term of the predictor, equation (16) is:

$$\begin{aligned} S_i = \frac{\Delta t}{\left(J^{-1}\right)^{n+1}}\Bigg\{ & \frac{3}{2}\left(C_i^n + D_E^n\left[u_i^n\right]\right) \\ & - \frac{1}{2}\left(C_i^{n-1} + D_E^{n-1}\left[u_i^{n-1}\right]\right) \\ & + \frac{1}{2}\left(D_I^n\left[u_i^n\right] + D_I^{n+1}\left[u_i^n\right]\right) \\ & + Q_i + \left(J^n - J^{n+1}\right)u_i^n\Bigg\} \end{aligned} \tag{21}$$

The use of time $(n+1)$ metric terms in the source of the predictor is allowed because our numerical method (see §3) solves for the time $(n+1)$ free surface position and the curvilinear grid prior to the solution of the predictor step.

Discrete operators from equations (16), (18), (20), and (21) are defined as:

$$D_\alpha\left(\,\right) = \frac{\Delta t}{2J^{-1}}\frac{\delta}{\delta\xi^\alpha}\left\{\nu J^{-1}g^{\alpha\alpha}\frac{\delta}{\delta\xi^\alpha}\left(\,\right)\right\} \tag{22}$$

where $\alpha = 1, 2, 3$ with no summation.

$$D_E\left(\,\right) = \frac{\delta}{\delta\xi^q}\left\{\nu J^{-1}g^{qr}\frac{\delta}{\delta\xi^r}\left(\,\right)\right\}_{q\neq r} \tag{23}$$

$$D_I\left(\,\right) = \frac{\delta}{\delta\xi^q}\left\{\nu J^{-1}g^{qr}\frac{\delta}{\delta\xi^r}\left(\,\right)\right\}_{q=r} \tag{24}$$

$$C_i = -\frac{\delta}{\delta\xi^q}\left\{J^{-1}U^q u_i\right\} \tag{25}$$

$$B_i\left(\,\right) = -\frac{\delta}{\delta\xi^q}\left\{J^{-1}\frac{\delta\xi^q}{\delta x_i}\left(\,\right)\right\} \tag{26}$$

$$Q_i = \frac{\delta}{\delta\xi^q}\left\{\left(J^{-1}\dot{X}^q\right)^{n+\frac{1}{2}}u_i^n\right\} \tag{27}$$

In the operator Q_i, we use equation (14) to define:

$$\left(J^{-1}\dot{X}^q\right)^{n+\frac{1}{2}} = \frac{1}{2\Delta t}\left\{J^{-1}\frac{\delta\xi^q}{\delta x_j}\bigg|^n + J^{-1}\frac{\delta\xi^q}{\delta x_j}\bigg|^{n+1}\right\}\left(x_j^{n+1} - x_j^n\right) \tag{28}$$

Note that in this method, the effect of the moving grid is carried in the grid velocity term (Q_i) which is the net contravariant flux of physical space through the sides of a control volume cell as viewed from computational space. The grid flux is a part of the source term for the computation of the intermediate (u^*) velocity, but does not explicitly appear in the pressure Poisson equation or the corrector steps. Therefore, the Poisson solver and corrector steps are only indirectly affected by the moving grid.

2.5 Kinematic boundary condition

The kinematic boundary condition is the Lagrangian condition that a particle on the surface must remain on the surface. It is possible to use the Lagrangian condition directly and advance the free surface by moving marker particles based upon their velocity at the free surface; however, this method has been shown to be unstable

in long simulations where the free-surface is advanced explicitly [20]. For fully-implicit solutions where the kinematic boundary condition is coupled in an implicit solution of the velocity, such instability should not occur with the Lagrangian boundary condition.

As an alternative to the Lagrangian approach, the kinematic boundary condition in physical space can be written in a physical space Eulerian form which can be obtained through a Taylor series expansion [2]:

$$\frac{\partial H}{\partial t} \;=\; u_3 \;-\; u_1 \frac{\partial H}{\partial x} \;-\; u_2 \frac{\partial H}{\partial y} \qquad (29)$$

where H is the height of the free surface measured from some horizontal baseline in physical space This form of the kinematic boundary condition is enforced on surface particles that are restricted to vertical motion in physical space, and is ideally suited to restricted curvilinear coordinate applications.

It has been demonstrated [12] that curvilinear coordinate transformations (equations (1) and (2)) can be applied to the physical space Eulerian kinematic boundary condition (equation (29)) for use in numerical simulations. The simulations that have used this approach [6] [12] have retained the underlying vertical motion restriction on surface particles, thereby making the methods unsuitable for waves which do not remain single-valued.

A more general approach that does not have a single-valuedness restriction in physical space requires deriving the Eulerian kinematic boundary condition directly in curvilinear coordinates. We have not seen this form of the kinematic boundary condition used by any previous authors and, therefore, present the derivation here even though it is, in some senses, a trivial extension of the textbook derivation of the physical space Eulerian kinematic boundary condition by Mei [2].

To directly obtain a curvilinear Eulerian kinematic boundary condition, we will consider a *fixed* curvilinear space $\left(\xi^1, \xi^2, \xi^3\right)$ such that the free-surface is single valued in ξ^3. Note that we are not requiring the curvilinear coordinate system to be boundary-fitted or moving for this derivation. Define F as a scalar function for the free surface such that:

$$F\left(\,\boldsymbol{\xi}\,,\,t\right) \;=\; \xi^3 \;-\; \mathcal{H}\left(\xi^1\,,\,\xi^2\,,\,t\right) \;=\; 0 \qquad (30)$$

where $\boldsymbol{\xi}$ is a vector representing the curvilinear coordinates of a surface position at time t, and $\mathcal{H}$ is the height of the free surface measured from $\xi^3 = 0$ along a line of constant ξ^1 and ξ^2 in fixed curvilinear space. After some small time Δt, the free surface has moved, while the curvilinear coordinate system remains fixed. We require that Δt is small, so the free surface remains single valued in ξ^3. Therefore,

$$\begin{aligned} F\left(\boldsymbol{\xi} \;+\mathbf{U}\Delta t\,,\,t+\Delta t\right) & \\ \;=\; F\left(\boldsymbol{\xi}\;,\,t\right) \;+\; \left(\frac{\partial F}{\partial t} \;+\; \mathbf{U}\cdot\nabla F\right)\Delta t & \\ +O\left(\Delta t\right)^2 & \qquad (31) \end{aligned}$$

where $\mathbf{U}$ is the contravariant vector velocity of a point on the surface. It follows that:

$$\frac{\partial F}{\partial t} \;+\; \mathbf{U}\cdot\nabla F \;=\; 0 \qquad (32)$$

Substitution of equation (30) provides the curvilinear kinematic boundary condition in *fixed* curvilinear coordinates as:

$$\frac{\partial \mathcal{H}}{\partial t} \;=\; U^3 \;-\; U^1 \frac{\partial \mathcal{H}}{\partial \xi^1} \;-\; U^2 \frac{\partial \mathcal{H}}{\partial \xi^2} \qquad (33)$$

To discretize the kinematic boundary condition, equation (33), we use a time-splitting implicit method that applies a Crank-Nicolson discretization for the surface position [20]:

$$\begin{aligned} \mathcal{H}^{[n+1]} \;-\; \mathcal{H}^{[n]} \;= & \\ \Delta t\left\{U^{3^{[n]}} \;-\; \frac{U^{q^{[n]}}}{2}\left(\frac{\delta\mathcal{H}}{\delta\xi^q}^{[n]} \;+\; \frac{\delta\mathcal{H}}{\delta\xi^q}^{[n+1]}\right)\right\} & \\ +O\left(\Delta t\right)^2 \qquad :\; q \;=\; 1,2 & \qquad (34) \end{aligned}$$

Now, we require that the *fixed* curvilinear grid used in the above derivation be boundary-fitted to the time (n) free surface; that is, at the surface:

$$\mathcal{H}^{[n]} \;=\; \xi^{3^{[n]}}\Big|_{surface} \;=\; \text{constant} \qquad (35)$$

then the gradients of the time (n) free-surface height relative to the ξ^1 and ξ^2 curvilinear coordinates will disappear, since:

$$\frac{\delta\mathcal{H}}{\delta\xi^q}^{[n]} \;=\; \frac{\delta\xi^3}{\delta\xi^q}^{[n]} \;=\; 0 \qquad :\; q \;=\; 1,2 \qquad (36)$$

An approximate-factorization of equation (34) using equation (36) provides our discrete kinematic boundary condition,

$$\left(I + \frac{\Delta t}{2} U^1 \frac{\partial}{\partial \xi^1}\right)\left(I + \frac{\Delta t}{2} U^2 \frac{\partial}{\partial \xi^2}\right)\left(\mathcal{H}^{n+1} - \mathcal{H}^n\right)$$

$$= \Delta t \left(U^3\right)^n + O\left(\Delta t\right)^2 \qquad (37)$$

Solution of equation (37) in three dimensions requires the inversion of two tridiagonal matrices to obtain $\mathcal{H}^{n+1}$. To obtain the time $(n+1)$ physical space position of a particle on the surface, we can use:

$$\Delta x_i = \frac{\partial x_i}{\partial \xi^3} \Delta \xi^3 \quad : \quad i = 1, 3 \qquad (38)$$

with equation (35) this can be written as:

$$x_i^{n+1} = x_i^n + \left(\mathcal{H}^{n+1} - \mathcal{H}^n\right)\left(\frac{\partial x_i}{\partial \xi^3}\right)^n \qquad (39)$$

Our method requires that at time (n) there is a fixed curvilinear grid that is boundary-fitted in the ξ^3 coordinate. To find the change in the ξ^3 coordinates of the free surface at some small time later with reference to the same fixed grid, we apply equation (37). Then equation (39) is used to obtain the physical space coordinates of the free surface. The new free surface can then be used to generate a new boundary-fitted grid. The advantage of this method over a curvilinear transformation of (29) is that equation (33) is enforced upon points which move along a line of constant ξ^1 and ξ^2 curvilinear coordinates rather than a line of constant x and y physical coordinates. As a result, the single-valuedness requirement in physical space is replaced by a single-valuedness requirement in curvilinear space, which is a less restrictive condition for a boundary-fitted coordinate system.

2.6 Dynamic boundary condition

The full equation for the dynamic boundary condition is quite complicated, and can be found in Scriven [28] and Aris [21]. If we neglect surface tension and its gradients, inertia of the surface, gradients of the dilational force, force due to total curvature and velocity, effects of varying normal velocity, normal forces due to dilation and shear, and the viscosity of the upper fluid, then we can write the dynamic boundary condition for an incompressible fluid in its classic form (similar to that in Batchelor [29]):

$$P_{S+} - P_{S-} = -2\mu e_{ij} n_i n_j \qquad (40)$$

$$e_{ij} t_i n_j = 0 \qquad (41)$$

where the subscripts $S+$ and $S-$ indicate the pressure on the upper and lower sides of the free surface, e_{ij} is the rate-of-strain tensor, and n_i and t_i are the unit normal and tangent vectors, respectively.

Equations (40) and (41) are a form of the dynamic boundary condition that does not provide for straightforward implementation in a boundary-fitted curvilinear coordinate numerical method; therefore, our approach will begin with the tensor form of the full equation from Scriven [28]. By applying the same simplifications used to get equations (40) and (41), along with the definition of the reduced pressure (equation (6)) and the requirement that the curvilinear coordinate system be boundary orthogonal, the dynamic boundary condition can be presented as:

$$\left(p_{S+} - p_{S-}\right) - g\left(Z_{S+} - Z_{S-}\right) = -2\nu U^3_{,3} \qquad (42)$$

$$U^1_{,3} = -g_{33}\left\{g^{11} U^3_{,1} + g^{12} U^3_{,2}\right\} \qquad (43)$$

$$U^2_{,3} = -g_{33}\left\{g^{22} U^3_{,2} + g^{12} U^3_{,1}\right\} \qquad (44)$$

where Z_{S+} and Z_{S-} are the vertical physical space coordinates on either side of the free surface along a curvilinear coordinate line that is normal to the free surface. Defining Δs as the thickness of the free surface, it follows that:

$$Z_{S+} - Z_{S-} \leq \Delta s \qquad (45)$$

In order to neglect the thickness of the free surface while maintaining second order accuracy in space, we require:

$$\Delta s \leq \left(\Delta x\right)^2 \qquad (46)$$

where Δx is the grid spacing (measured in physical space) at the free surface. Equations (45) and (46) allow equation (42) to be written as:

$$\left(p_{S+} - p_{S-}\right) = -2\nu U^3_{,3} + O\left(\Delta x\right)^2 \qquad (47)$$

Note that the differentiation in equations (43), (44), and (47) is covariant tensor differentiation and requires the application of Christoffel symbols for deriving a discrete implementation.

In order to get the reduced pressure (p) from equation (47) into terms of the pressure variable (ϕ), we multiply equation (20) by $\delta \xi^r / \delta x_i$, with a sum over $i = 1, 2, 3$; then use equations (7), (15), (24) and (26) to obtain:

$$J^{-1}g^{qr}\frac{\delta p}{\delta\xi^q} \;=\; J^{-1}g^{qr}\frac{\delta\phi}{\delta\xi^q}$$

$$-\frac{\Delta t}{2}\frac{\delta\xi^r}{\delta x_i}\frac{\delta}{\delta\xi^q}\left\{\nu J^{-1}g^{qs}\frac{\delta}{\delta\xi^s}\left(\frac{\delta\xi^t}{\delta x_i}\frac{\delta\phi}{\delta\xi^t}\right)\right\} \quad (48)$$

Let $r = 3$ and apply boundary orthgonality so that $g^{13} = g^{23} = 0$, then equation (48) can be reduced to:

$$\frac{\delta p}{\delta\xi^3} \;=\; \frac{\delta\phi}{\delta\xi^3} \;-\; O\left(\Delta t\right) \quad (49)$$

A discrete version of equation (49) across a free surface of thickness $\left(\xi^3_{S+} - \xi^3_{S-}\right)$, in curvilinear space, can be written as

$$\begin{aligned} p_{S+} \;-\; p_{S-} \;&=\; \phi_{S+} \;-\; \phi_{S-} \\ &-\; O\left(\Delta t\right)\left(\xi^3_{S+} \;-\; \xi^3_{S-}\right) \end{aligned} \quad (50)$$

The relationship between physical and curvilinear coordinates, equation (38), provides:

$$Z_{S+} - Z_{S-} \;=\; \frac{\partial x_3}{\partial\xi^q}\left(\xi^q_{S+} \;-\; \xi^q_{S-}\right) \quad (51)$$

where Z_{S+} and Z_{S-} are the vertical physical space coordinates on either side of the free surface along a curvilinear coordinate line that is normal to the free surface. Our grid spacing (in physical space) is defined as Δx while the grid spacing in curvilinear space is 1, so we can write:

$$Z_{S+} - Z_{S-} \;=\; \Delta x\left(\xi^3_{S+} \;-\; \xi^3_{S-}\right) \quad (52)$$

Using equations (45) and (46), it follows that,

$$\Delta x \;\geq\; \xi^3_{S+} \;-\; \xi^3_{S-} \quad (53)$$

Because our numerical method uses an explicit discretization (Adams-Bashforth) for the convective terms, we are subject to the Courant-Friedrichs-Lewy condition:

$$\frac{u\Delta t}{\Delta x} \;<\; 1 \quad (54)$$

So, if we apply equations (53) and (54) to equation (50), the result is second-order accurate in space:

$$p_{S+} \;-\; p_{S-} \;=\; \phi_{S+} \;-\; \phi_{S-} \;-\; O\left(\Delta x\right)^2 \quad (55)$$

Applying equation (55) and some algebra and tensor manipulation, the dynamic boundary condition of equations (43), (44), and (47) can be reduced to a form that can be more readily implemented in a numerical method:

$$\begin{aligned} \frac{\partial U^3}{\partial\xi^3} \;&=\; -\;\frac{1}{2\mu}\left(\phi_{S+} \;-\; \phi_{S-}\right) \\ &-\; \frac{1}{2}g^{33}\left(U^1\frac{\partial g_{33}}{\partial\xi^1} \;+\; U^2\frac{\partial g_{33}}{\partial\xi^2}\right) \end{aligned} \quad (56)$$

$$\frac{\partial U^1}{\partial\xi^3} \;=\; -g_{33}\left\{g^{11}\frac{\partial U^3}{\partial\xi^1} \;+\; g^{12}\frac{\partial U^3}{\partial\xi^2}\right\} \quad (57)$$

$$\frac{\partial U^2}{\partial\xi^3} \;=\; -g_{33}\left\{g^{22}\frac{\partial U^3}{\partial\xi^2} \;+\; g^{12}\frac{\partial U^3}{\partial\xi^1}\right\} \quad (58)$$

3 NUMERICAL METHOD

We have expanded the non-staggered grid, approximate-factorization, time-splitting method of Zang [19] to handle moving grids and a free surface. Previous authors [22] [27] have noted that appropriate discretization of the Jacobian in a moving grid method is important in maintaining numerical accuracy. In our method, the Jacobian is updated using the conservation of space, equation (8). By applying the definition of the grid flux, $\dot{X}$, equation (14), with an explicit Euler discretization, the conservation of space can be written as:

$$\left(J^{-1}\right)^{n+1} \;=\; \left(J^{-1}\right)^n \;+\; \Delta t\frac{\delta}{\delta\xi^q}\left(J^{-1}\dot{X}^q\right)^{n+\frac{1}{2}} \quad (59)$$

Equation (59) requires that the change in the physical-space volume contained in a computational space cell must be computed by summing the fluxes of physical space through the faces of the computational space cell. This ensures that space is numerically conserved and prevents an inconsistency between the Jacobian computation and the convective grid flux term, equation (27). To demonstrate that our moving grid method is second-order accurate in both space and time, numerical simulations of a decaying vortex (without a free surface) have been conducted and are reported in §5.1.

The numerical method can be summarized as:

1. Use the kinematic boundary condition to advance the free surface from time (n) to time $(n+1)$.

2. Compute grid and metrics for time $(n+1)$ grid using algebraic or Poisson solution method.
3. Solve for u_i^* at center of cells.
4. Use quadratic interpolation (QUICK [30]) to obtain the normal component of U^* on each cell face.
5. Use surface decomposition to solve the Poisson pressure equation for the pressure variable ϕ, coupled with the dynamic boundary condition (which solves for the contravariant velocity U^q on the free surface and the pressure variable ϕ_{S-} at the free surface).
6. Compute time $(n+1)$ Cartesian velocity (u_i) at cell centers.
7. Compute time $(n+1)$ contravariant velocity components normal to cell surfaces, U^q.
8. Compute all three components of contravariant velocity U^q on free surface

The free surface advance (step 1) and the u^* computation (step 3) are straightforward numerical implementations of equations (37) and (16) using vectorized tridiagonal solvers. The grid generation (step 2) uses the EAGLE code discussed in §2.2. The surface decomposition solution of the Poisson pressure equation and the dynamic boundary condition (step 5) is the most interesting part of our method and is covered in-depth in §4. Applying the corrector equations (steps 6 and 7) provides the three components of Cartesian velocity at the cell centers in the fluid volume, and one contravariant flux at each cell edge by solving equations (18) and (19). In order to advance the free surface using the kinematic boundary condition in the next time step, it is necessary to compute all three contravariant velocity components on the free surface (step 8). This can be done by using the dynamic boundary condition, equations (56), (57), and (58), and the contravariant velocity normal to the free surface (computed in the surface decomposition).

4 SURFACE DECOMPOSITION

Inherently, the invoking of the dynamic boundary condition presents difficulties for a Poisson solver due to the complicated interrelation between the boundary pressure and velocity gradients that is found in equations (56), (57), and (58). It is possible to directly enforce the dynamic boundary condition in the relaxation of the Poisson pressure equation; however, this is likely to impact future utility of the code. The difficulties involved can be illustrated by considering that the dynamic boundary condition directly affects only the layer of control volumes adjacent to the surface; however, in a multigrid method, this influence is moved progressively further into the domain as the grid is coarsened. This presents challenges in the derivation and coding of the restriction and interpolation operators that are used for mesh coarsening and refinement. Our investigations lead us to believe that this will make it difficult to develop future refinements of the dynamic boundary condition with reasonable amounts of effort.

To get around this problem, we have adapted the domain decomposition methods of Zang [31] to split the solution of the dynamic boundary condition from the solution of the pressure Poisson equation. That is, we decompose our solution into a two-dimensional surface and a three-dimensional volume for the pressure solution step. In Zang's method (as originally developed), the pressure Poisson equation is relaxed iteratively through decomposed domains using one or two V-cycles of the multigrid solver on each domain. Pressure and velocity boundary conditions are exchanged only at the finest multigrid level. The result is a method that quickly converges to a consistent pressure field over decomposed domains. Our new adaptation manipulates the dynamic boundary condition into an equation for the pressure at the free-surface and an equation for the contravariant flux across the free-surface. Therefore, we can alternately sweep the free-surface and the volume in a manner similar to that presented by Zang [31]. Our information exchange at the free-surface provides the pressure Poisson equation with a Dirichlet contravariant-velocity boundary condition without requiring any changes in the restriction and prolongation operators. This method is more suitable for future expansions since the dynamic boundary condition can be changed by working only with the surface domain, without affecting the volume domain and the fluid-flow Poisson solver.

In addition to those already presented, two other types of equations are required to obtain our discrete implementation of the dynamic boundary condition. One is a discrete representation of the contravariant velocity on the free

surface based upon the gradient of the contravariant velocity at the free surface and below the free surface:

$$\left.U^{*q}\right|_S = \left.U^{*q}\right|_{S-1} + \frac{1}{4}\left.\frac{\delta}{\delta\xi^3}U^{*q}\right|_S + \frac{1}{4}\left.\frac{\delta}{\delta\xi^3}U^{*q}\right|_{S-1} \quad (60)$$

where subscript (S) indicates the discrete quantities are evaluated at the cell face on the free surface, and subscript $(S-1)$ indicates the quantities are evaluated at the cell center inside the free surface.

The second type of equation required for completeness is an interpolating method for computing the first and second derivatives of the pressure variable (ϕ) at the free surface from the pressures at and below the free surface. For a two-dimensional flow with a one-dimensional surface we have tested both linear and quadratic interpolating forms, which can be written in the general form:

$$\gamma \equiv \left.\frac{\delta\phi}{\delta\xi^3}\right|_S = \alpha_1\phi_{S-} + \alpha_2\phi_{S-1} + \alpha_3\phi_{S-2} \quad (61)$$

$$\psi \equiv \left.\frac{\delta^2\phi}{\left(\delta\xi^3\right)^2}\right|_S = \alpha_4\phi_{S-} + \alpha_5\phi_{S-1} + \alpha_6\phi_{S-2} \quad (62)$$

where quantities subscripted with S and $S-1$ are as noted above, while ϕ_{S-} indicates the pressure in the water at the free surface, and ϕ_{S-2} indicates the pressure at the center of the second cell center inside the free surface.

The derivation of the three-dimensional discretized equations to be enforced as the dynamic boundary condition in the two-dimensional surface domain is too long to present in this paper. The method requires that equations (56), (57), (58), (60), (61), and (62), be manipulated to express the dynamic boundary condition as two linear operators: one for the contravariant velocity component normal to the free surface,

$$L_1\left(U_S^3\right) = S_1\left(\phi_{S-}, \phi_{S-1}, U_{S-1}^*, \gamma, \psi\right) \quad (63)$$

and a second operator for the pressure in the water at the free surface,

$$L_2\left(\phi_{S-}\right) = S_2\left(\phi_{S+}, U_{S-1}^*, U_S^3, \gamma\right) \quad (64)$$

where ϕ_{S+} indicates the pressure in the air at the free surface.

For a three-dimensional flow with a two-dimensional free surface, the linear operators L_1 and L_2 take a discretized Poisson-like form and are solvable by multigrid methods. For the two-dimensional flows with one-dimensional surfaces that we have tested to date, the linear operators take on tridiagonal form and are solved by inversion. The coupled two-dimensional surface decomposition method can be summarized as:

1. Solve the free-surface pressure equation (64), for an estimated free-surface pressure (ϕ_{S-}), using the pressure gradient and contravariant velocity from the last iteration.
2. Solve the free surface contravariant velocity equation (63) for an estimated contravariant velocity $\left(U_S^3\right)$ at the surface.
3. Repeat steps 1 and 2.
4. Perform one multigrid V-cycle, relaxing equation (17) to obtain a pressure estimate in volume.
5. Compute the estimated free-surface pressure gradient, γ, and second derivative of pressure, ψ, from equations (61) and (62) respectively.
6. Repeat steps 1 through 5 until convergence.

We expect that the three-dimensional implementation of this method (currently under development) will involve two V-cycles of the free surface pressure and contravariant velocity multigrid equations for each V-cycle of the Poisson pressure equation. This method is likely to be more costly in computational time than a direct application of the dynamic boundary condition, but the overall computational costs should still be reasonable since they should be less than that for a three-dimensional, two-grid, domain decomposition case, such as Zang [31] has already shown to be feasible.

5 SIMULATION RESULTS

5.1 Decaying Vortex with a Moving Grid

The use of second-order accurate discretizations does not guarantee second-order accuracy in a numerical simulation [32]. This is especially true

with boundary-fitted curvilinear coordinates and moving grids. We have conducted several simulations of a decaying vortex to demonstrate that our Navier-Stokes solution method is second-order accurate in time and space with a moving grid, The decaying vortex is an analytical solution of the two-dimensional Navier-Stokes equations over the domain of $(0 \leq x_1, x_2 \leq \pi)$ that can be written as:

$$u_1 = -\cos(x_1)\sin(x_2)\, e^{-2t} \quad (65)$$

$$u_2 = \sin(x_1)\cos(x_2)\, e^{-2t} \quad (66)$$

$$p = -0.25\left[\cos 2x_1 + \cos 2x_2\right] e^{-4t} \quad (67)$$

Figure (3) provides simulation results showing the reduction of the RMS velocity error with the increase in grid points from 8x8, to 16x16, to 32x32. Three different error lines are shown: the first represents the results for a fixed grid; the second is for a grid that is fixed in shape, but translates through the decaying vortex domain; the third is for a grid that has boundaries which remain fixed, but whose interior grid lines are stretched with each time step. It can be seen that accuracy is approximately second-order in all these cases.

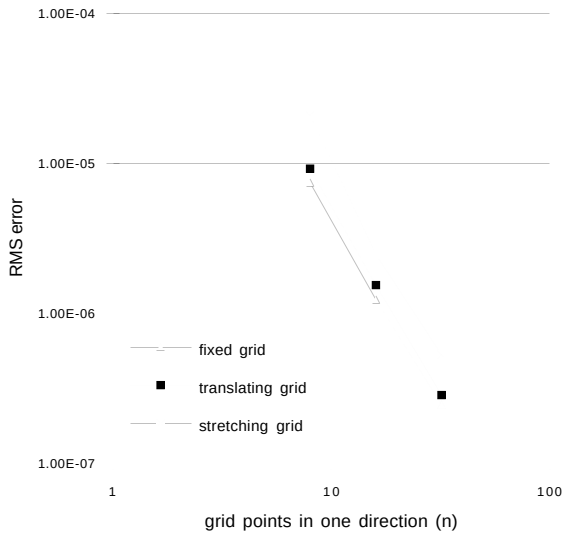

Figure 3: Decaying vortex accuracy

5.2 Standing Waves

To test the ability of our code to simulate a free surface, we performed simulations of standing waves in a two-dimensional rectangular basin. Free-slip boundary conditions were used for the sides and bottom of the 32 x 32 cell domain so that the simulations could be compared to nonlinear and linear standing wave theory. The grid points were distributed evenly in both directions. Using the wavelength $L = 1$ to non-dimensionalize the domain, the width is 0.5, and the depth d is 0.5. Simulations were run for small-amplitude waves with a non-dimensional wave height (or steepness) of 0.01 and for finite-amplitude waves with a non-dimensional wave height of 0.1. Using the wave amplitude a and the wave number k, these cases have ak values of 0.031 and 0.31, respectively. We conducted simulations at Reynolds numbers of 10, 100, and 1000 to illustrate capabilities of the simulation code with the relatively coarse grid. The Reynolds number is defined as:

$$Re = \frac{L\mathcal{U}}{\nu} \quad (68)$$

with ν as the kinematic viscosity, and $\mathcal{U}$ as the characteristic Cartesian velocity based upon the wave amplitude (a) and the wave frequency (ω):

$$\mathcal{U} = a\omega \quad (69)$$

From Lamb [33], the damping of a free wave due to viscosity as a function of time can be approximated from:

$$a(t) = a(0)e^{-2\nu k^2 t} \quad (70)$$

Note that this is based upon an energy dissipation argument for linear waves in deep-water, so we can only expect this to provide a rough guide to our expected damping.

From Wiegel [34] we can obtain predictions of wave period and shape for linear and nonlinear standing waves. Both linear and non-linear irrotational wave theory predict the wave period (T) for an inviscid wave as:

$$T = \sqrt{\frac{2\pi L}{g}\left(\tanh\frac{2\pi d}{L}\right)^{-1}} \quad (71)$$

For our simulation domain this provides a theoretical period of 1.1339 seconds.

According to linear theory for small amplitude waves, the wave shape should be a sinusoid, where the surface height (H) above the still water level is:

$$H(x,t) = a\sin(kx)\sin(\omega t) \quad (72)$$

where ω is the wave frequency ($2\pi/T$). Nonlinear theory for finite amplitude standing waves predicts a wave shape given by:

case	ak	Re	initial wave shape	comparison of simulation and theory: period difference[1]	wave height difference[2]	linear wave shape difference[3]	nonlinear wave shape difference[3]
1a	0.031	10	linear	0.32 %	1.1 %	0.39 %	0.32 %
1b	0.031	10	non-linear	0.32 %	1.1 %	0.37 %	0.31 %
2a	0.031	100	linear	0.23 %	0.76 %	1.5 %	0.54 %
2b	0.031	100	non-linear	0.23 %	0.64 %	1.1 %	0.22 %
3a	0.31	10	linear	11 %	0.59 %	0.33 %	0.20 %
3b	0.31	10	non-linear	11 %	0.64 %	0.33 %	0.12 %
4a	0.31	100	linear	0.87 %	1.9 %	2.4 %	0.59 %
4b	0.31	100	non-linear	0.87 %	1.1 %	2.2 %	0.37 %
5a	0.31	1000	linear	0.32 %	6.4 %	1.2 %	0.53 %
5b	0.31	1000	non-linear	0.87 %	3.7 %	1.2 %	0.32 %

NOTES:

1. "period difference" is mean difference between the simulation wave period and theoretical period for the four oscillations simulated, and is expressed as a percentage of the theoretical period.

2. "wave height difference" is RMS difference between the simulation and theoretical wave height for crests at $x = 0$ from the first through fourth periods, and is expressed as a percentage of the wave height.

3. the "wave shape difference" is the RMS difference between the simulation wave shape and theory for one wave, expressed as a percentage of the wave height, and measured at the second wave period for case 3 and the fourth wave period for all other cases.

Table 1: Simulation Summary

$$H(x,t) = a\sin(kx)\sin(\omega t) - \frac{1}{2}ka^2\coth(kd)\cos(2kx)\left\{\sin^2(\omega t) - \frac{3\cos(2\omega t) + \tanh^2(kd)}{4\sinh^2(kd)}\right\} \quad (73)$$

Note that for a small amplitude standing wave, the wave shape predicted by linear and non-linear theory are practically indistiguishable.

We have run simulations using both the linear and non-linear free-surface shape as an inital condition in order to examine the ability of the method to move toward the correct free-surface shape. A summary of the simulation conditions and comparisons between simulation results and theory are presented in table (1). Note that the results show excellent agreement for both the nonlinear wave shape and viscous damping of the wave height. In all cases, the wave shape is closer to the theoretical nonlinear shape than to the linear shape. The wave shape difference was computed independently of the effects of viscous damping by using the simulated wave amplitude in equations (72) and (73) instead of the theoretical amplitude from equation (70). Similarly, an adjustment was made for the difference between the theoretical period and the simulation period by applying a small time shift to adjust the theoretical crest to the same time as the simulation crest. The simulations show an increase in the simulation wave period with increasing viscosity, which is a realistic physical result.

To provide a better picture of the simulation accuracy, we present two types of graphs which compare the results of typical simulations to linear and nonlinear theory. Figures (4), (6), (8), (10), and (12) show the height of the free surface at the wall $x = 0$ as a function of non-dimensional time (which is obtained using the theoretical period computed from equation (71)). The lines for linear and nonlinear theory are based on equations (72) and (73) with equation (70) used to compute the theoretical amplitude as a function of time. Figures (5), (7), (9), (11), (13), and (14) compare the wave shape for the simulation wave and theory.

Figures (4) and (5) present results for case 1b, showing that the simulation of a small amplitude standing wave at a low Reynolds number maintains the correct wave profile and is damped as predicted by theory. The results for case 1a (using a linear initial wave shape), are indistinguishable from the results from case 1b (using the non-linear initial wave shape). Figures (6) and (7) present results for the small amplitude standing wave at a Reynolds number of 100, case 2b. These graphs demonstrate that the simulation of viscous, small-amplitude wave still

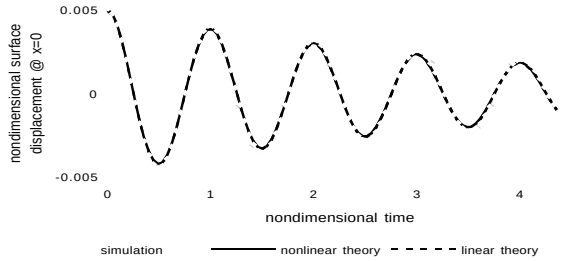

Figure 4: Free surface wall height, case 1b; $Re = 10$, $ak = 0.031$

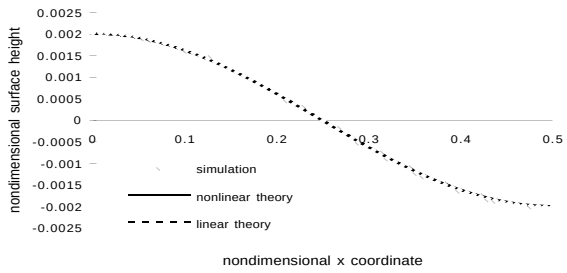

Figure 5: Free surface wave shape, case 1b; $Re = 10$, $ak = 0.031$

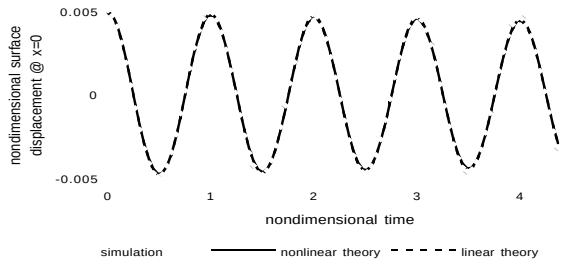

Figure 6: Free surface wall height, case 2b; $Re = 100$, $ak = 0.031$

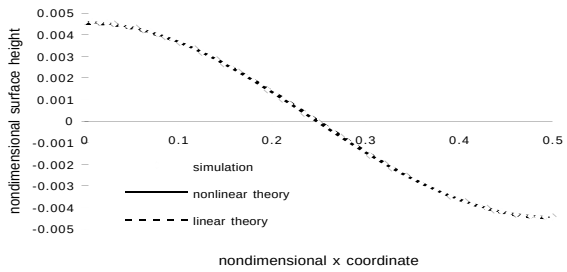

Figure 7: Free surface wave shape, case2b; $Re = 100$, $ak = 0.031$

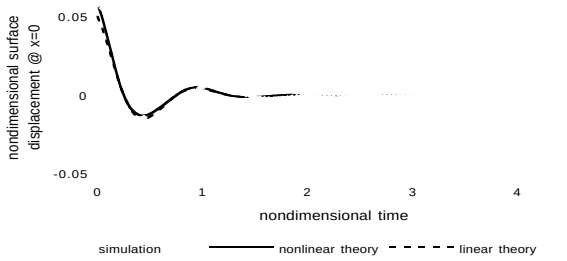

Figure 8: Free surface wall height, case 3b; $Re = 10$, $ak = 0.31$

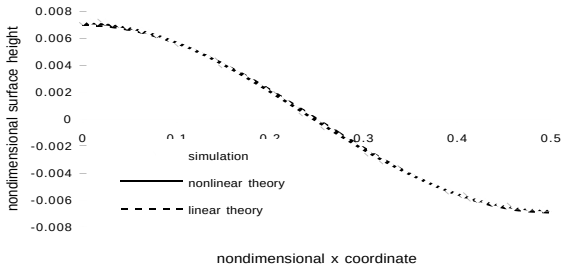

Figure 9: Free surface wave shape, case 3b; $Re = 10$, $ak = 0.31$

follows the linear and nonlinear theory both in wave shape and in period.

Figures (8) and (9) present results for the finite amplitude standing wave at $Re = 10$, case 3b. In this case, the wave is rapidly damped out. It can be seen that the period for the simulation is significantly greater than that for the theory. This result is not unreasonable, as one would expect that a highly viscous flow will oscillate at a slower period than that predicted by inviscid theory.

Figures (10) and (11) present results for the finite amplitude standing wave at $Re = 100$ with an initial nonlinear wave shape, case 4b. It can be seen that linear and non-linear theory are not coincident and the simulation wave shape follows nonlinear theory very closely.

Figures (12) and (13) present results for case 5b ($Re = 1000$) for an initially nonlinear wave shape. It can be seen that this case is at the limit of our ability to resolve the viscous effects with the coarse grid used in the simulation.

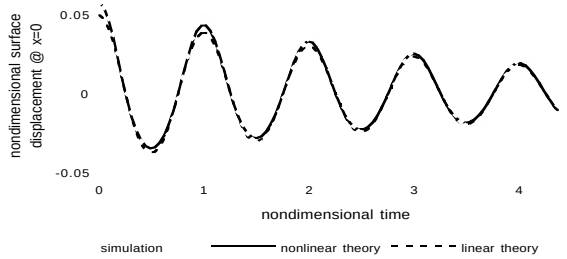

Figure 10: Free surface wall height, case 4b; $Re = 100$, $ak = 0.31$

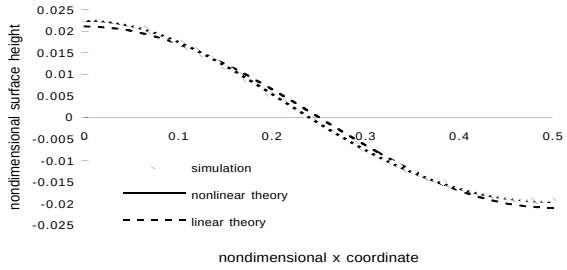

Figure 11: Free surface wave shape, case 4b; $Re = 100$, $ak = 0.31$

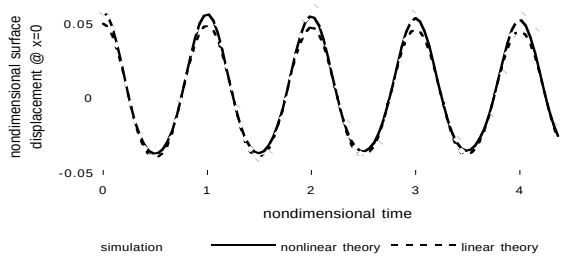

Figure 12: Free surface wall height, case 5b; $Re = 1000$, $ak = 0.31$

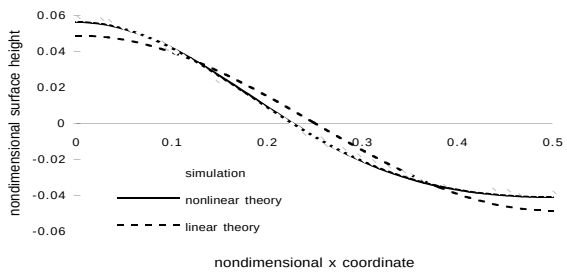

Figure 13: Free surface wave shape, case 5b; $Re = 1000$, $ak = 0.31$

To show that the simulation can evolve a nonlinear wave from an initially linear profile, figure (14) presents the wave shape for case 4a, which is similar to case 4b, (shown in figure (11)), except that the initial wave shape is a sinusoid from linear theory.

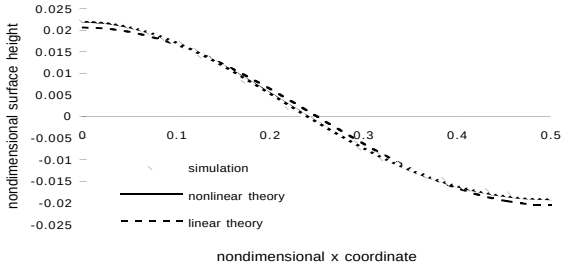

Figure 14: Free surface wave shape, case 4a; $Re = 100$, $ak = 0.31$

CONCLUSION

The numerical method presented has been shown to be second-order accurate in time and space with a moving grid, and has been shown to effectively simulate two-dimensional nonlinear waves. The method presented for deriving the kinematic boundary condition does not artificially restrict the free surface movement or limit wave shapes to single-valued functions in physical space. The surface decomposition method presented has been shown to be effective in the simulation of two-dimensional non-linear waves, and it is expected that future three-dimensional implementation will provide an effective method for invoking extremely complicated implementations of the dynamic boundary condition.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Fluid Dynamic Program, Office of Naval Research under grant nos. N00014-91-J-1200 and N00014-94-J-0190.

References

[1] Floryan J.M., and Rasmussen, H., "Numerical methods for viscous flows with moving boundaries," *Appl. Mech. Rev.*, Vol. 42, 1989, pp. 323–340.

[2] Mei, C.C., *The Applied Dynamics of Ocean Surface Waves*. Wiley-Interscience, 1983.

[3] Harlow F.H., and Welch, J.E., "Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface," *Physics of Fluids*, Vol. 8, 1965, pp. 2182–2189.

[4] Hino, T. "Computation of viscous flows with free surface around an advancing ship," Presented at the second international colloquium on viscous fluid dynamics in ship and ocean technology, Osaka, Japan, September 1991.

[5] Thomas, T.G., and Leslie, D.C., "Development of a conservative 3-D free surface code," *J. Hydraulic Research*, Vol. 30, 1992, pp. 107–115.

[6] Park, J.-C., Zhu, M., and Miyata, H., "On the accuracy of numerical wave making techniques," *J. of The Society of Naval Architects of Japan*, Vol. 173, 1993, pp. 35–44.

[7] Sussman, M., Smereka, P., and Osher, S., "A level set approach for computing solutions to incompressible two-phase flow," Department of Mathematics, CAM Report 93-18, University of California, Los Angeles, June 1993.

[8] Zhu, J., and Sethian, J., "Projection methods coupled to level set interface techniques," *J. Comp. Physics*, Vol. 102, 1992, pp. 128–138.

[9] Hino, T., Martinelli, L., and Jameson, A., "A finite-volume method with unstructured grid for free surface flow simulations," Presented at the sixth international conference on numerical ship hydrodynamics, Ship Research Institute, Tokyo, Japan, August 1993.

[10] Loh, C.Y., and Rasmussen, H., "A numerical procedure for viscous free surface flows," *Appl. Numerical Mathematics*, Vol. 3, 1987, 479–495.

[11] Dommermuth. D.G., "The laminar interactions of a pair of vortex tubes with a free surface," *J. Fluid Mech.*, Vol. 246, 1993, 91–115.

[12] Hinatsu, M., "Numerical simulation of unsteady viscous nonlinear waves using moving grid system fitted on a free surface," *J. of Kansai Society of Naval Architects, Japan*, No. 217, 1992, pp. 1–12.

[13] Miyata, H., Zhu, M., and Watanabe, O., "Numerical study on a viscous flow with free-surface waves about a ship in steady straight course by a finite-volume method," *J. Ship Research*, Vol. 36, 1992, pp. 332–345.

[14] Ohring, S., and Lugt, H.J., "Two counter-rotating vortices approaching a free surface in a viscous fluid," Report 89/013, David Taylor Research Center, 1989.

[15] Farmer, J., Martinelli, L., and Jameson, A., "Multigrid solutions of the Euler and Navier-Stokes equations for a series 60 Cb = 0.6 ship hull for Froude numbers 0.160, 0.220 and 0.316 (program 1: Navier-Stokes formulation)," CFD Workshop Tokyo, Japan, March 1994.

[16] Wang, H.T., and Leighton, R.I., "Three-dimensional vortex interactions with a free surface," *Recent Advances and Applications in Computational Fluid Dynamics*, Vol. FED-103, ASME, 1990, pp. 213–222.

[17] Kassinos, A.C., and Prusa, J., "A numerical model for 3-D viscous sloshing in moving containers," *Recent Advances and Applications in Computational Fluid Dynamics*, Vol. FED-103, ASME, 1990, pp. 75–86.

[18] Zhan, C., and Zhaoshun, Z., "Numerical calculation of drift current in wind waves," *Acta Oceanologica Sinica*, Vol. 11, No. 2, 1992, pp. 179–188.

[19] Zang, Y., Street, R.L., and Koseff, J.R., "A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in general curvilinear coordinate systems," To appear in *J. Comput. Physics*, Aug. 1994.

[20] Chan, R.K.-C., and Street, R.L., "SUMMAC – A numerical model for water waves," Technical Report 135, Stanford University, Aug. 1970.

[21] Aris, R., *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover, New York, 1962.

[22] Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W., *Numerical Grid Generation*. Elsevier Science Publishing Co., New York, 1985.

[23] Zang, Y., *On the Development of Tools for the Simulation of Geophysical Flows.* PhD thesis, Stanford University, Dept. Mech. Eng., 1993.

[24] Thompson, J.F., "Current developments in grid generation for complex configurations," *Ocean Waves Mechanics, Computational Fluid Dynamics, and Mathematical Modelling*, Southampton, 1990, pp. 321–340.

[25] Perng, C.Y., *Adaptive-Multigrid Computations for Incompressible Flows, Including Geometry, Temperature, and Salinity Effects.* PhD thesis, Stanford University, Dept. Mech. Eng., 1990.

[26] Meakin, R.L., and Street, R.L., "Simulation of environmental flow problems in geometrically complex domains. Part I: A general coordinate transformation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 68, 1988, pp. 151–175.

[27] Demirdžić, I., and Perić, M., "Space conservation law in finite volume calculations of fluid flow," *International J. for Numerical Methods in Fluids*, Vol. 8, 1988, pp. 1037–1050.

[28] Scriven, I.E., "Dynamics of a fluid interface," *Chemical Engineering Science*, Vol. 12, 1960, pp. 98–108.

[29] Batchelor, G.K., *Fluid Dynamics.* Cambridge University Press, 1967.

[30] Leonard, B.P., "A stable and accurate convective modelling procedure based on quadratic upstream interpolation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, 1979, pp. 59–98.

[31] Zang, Y., Street, R.L., and Koseff, J.R., "A composite-multigrid method for solving unsteady incompressible Navier-Stokes equations in complex geometries," In *Numerical Methods in Laminar and Turbulent Flow*, Vol. VII, Pineridge Press, 1991, pp. 1485–1495.

[32] Kim, J., and Moin, P., "Application of a fractional-step method to incompressible Navier-Stokes equations," *J. Comp. Physics*, Vol. 59, 1985, pp. 308–323.

[33] Lamb, H., *Hydrodynamics.* Dover, 1945.

[34] Wiegel, R.L., *Oceanographical Engineering.* Prentice-Hall, Inc., 1964.