

Advances in Cartesian-grid methods for computational ship hydrodynamics

G. Weymouth¹, D.G. Dommermuth², K. Hendrickson¹, and D.K.-P. Yue¹
¹(Massachusetts Institute of Technology, USA)
²(Science Applications International Corporation, USA)

ABSTRACT

Recent developments in computational fluid dynamics are enabling accurate simulations of full scale ship hydrodynamic flow. In particular, Cartesian-grid techniques such as volume-of-fluid and immersed boundary methods expedite computational simulations of unsteady breaking wave flows around complicated ship geometries. However, there are still a number of important numerical and modeling issues to overcome such as finite computational domains, moving bodies, and dynamic slip conditions which are required in order for Cartesian-grid methods to achieve their full potential as a practical analysis tool. Additionally, systematic studies of the accuracy and applicability of numerical techniques are key for their use in quantitative engineering applications. In this work we develop a new method for the enforcement of general boundary conditions on an arbitrary body as well as a new exit condition to allow for significantly smaller domains and a volume of fluid algorithm to preserve the free surface characteristics. We consider a number of canonical problems to establish quantitative comparisons and assessments of these new methods. We use these new capabilities to preform flow simulations with naval applications and compare them to experimental results.

INTRODUCTION

Scientific and engineering investigation into ocean systems give rise to a wide variety of complex nonlinear fluid dynamics problems. Historically the solutions techniques used to solve these problems were of limited accuracy and applicable only to a very small subset of the complete system. Even with the continuing increase of availability of computational resources, producing quality solutions with modern computational fluid dynamic

techniques requires extensive time and training, limiting their use in practical science and engineering applications.

Cartesian-grid methods are a significant advancement in the simulation of general fluid flows because they afford the capability of computing flows with engineering applications without the limitations and difficulties associated with non-orthogonal or unstructured grids. Recently, Hendrickson and Yue (2006) show the level-set method is capable of high-resolution computations of unsteady small-scale breaking waves. Dommermuth et al. (2005) use the volume-of-fluid method to simulate full-scale ship breaking waves at high Froude numbers on Cartesian grids and Dommermuth et al. (2006) present high-resolution unsteady predictions of the free-surface elevation and wave making resistance of a model 5415 hull.

While the move to Cartesian-grid methods is a step in the correct direction, there are still many issues which have not received a simple and general treatment. These techniques must be comprehensive, robust, and well validated before Cartesian-grid methods will be useful in an engineering setting. This paper forwards such advancements in free surface tracking, exit boundary conditions and body boundary treatment. These new techniques are developed using a flow solver based on the algorithms described in Dommermuth et al. (2005) such that they will be easily incorporated into similar large-scale commercially ready codes. The techniques are verified with two and three-dimensional canonical simulations. They are then used to preform initial investigations into Cartesian-grid simulations of 2D+t bow waves. A wave maker test based on this methodology is used to validate the code and motivate discussion of practical models of tangential flow for Cartesian-grid methods.

PROBLEM FORMULATION AND GENERAL SOLVER DESCRIPTION

Our physical system is the general fluid flow near an air-water interface. For our model it is required that the fluid velocity \vec{u} satisfy the conservation of momentum for an incompressible inviscid fluid, given by the Euler equation as

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \vec{\nabla})\vec{u} = -\frac{1}{\rho}\vec{\nabla}p - \vec{g} \quad (1)$$

where p is the total pressure, \vec{g} is the gravitational acceleration vector, and ρ is the local fluid density. For convenience, the vector \vec{r} is defined as the combination of the convective and gravitational terms, such that (1) becomes

$$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho}\vec{\nabla}p + \vec{r}. \quad (2)$$

The effects of surface tension are ignored for the large-scale flows studied in this work.

Taking the divergence of (2) results in a variable coefficient Poisson equation for the pressure of the form

$$\vec{\nabla} \cdot \left(\frac{1}{\rho} \vec{\nabla} p \right) = \vec{\nabla} \cdot \left(\vec{r} - \frac{\partial \vec{u}}{\partial t} \right). \quad (3)$$

The pressure field resulting from the solution of this equation is used to project the velocity field onto one satisfying the divergence free constraint

$$\vec{\nabla} \cdot \vec{u} = 0. \quad (4)$$

We use a volume-of-fluid method to model free-surface flows. In this framework the local fluid density is calculated as

$$\rho = \rho_w f + \rho_a (1 - f) \quad (5)$$

where ρ_w and ρ_a are the densities of water and air respectively. The field f is the fraction of water filling each cell and our scheme for determining this field will be discussed in detail in the following section.

Our basic implementation of these equations follows the method of the Numerical Flow Analysis (NFA) code (Dommermuth et al. 2004). The discrete forms of equations (2) and (3) are posed on a Cartesian grid covering the fluid domain. Staggered variable placement is used. The time derivatives are treated with an explicit low storage second-order Runge-Kutta method (Dommermuth et al. 2004). The pressure terms

are treated conservatively using central differences and a preconditioned conjugate-gradient method is used to iteratively solve the Poisson equation. The convective terms in NFA are treated with a slope-limited QUICK scheme for stability and accuracy.

The NFA code is well documented as providing excellent high fidelity turnkey solutions to ship flows. However, there are still a number of challenging issues which must be overcome for Cartesian-grid methods to robustly simulate completely general two and three-dimensional free-surface flows. First, the free surface must be tracked accurately and consistently for simulated results to compare well with experiments. Existing methods for the advection of the volume fraction f describing a three-dimensional free surface are highly complex and do not conserve fluid volume at the interface. This can lead to free surface “stepping” as well as the generation of flotsam and jetsam. In the next section we present a list of requirements for an operator-split algorithm to conserve fluid volume to machine precision and present a simple dimensionally independent method which meets these requirements.

The second advancement which we investigate in this work is an improvement to exit boundary conditions. For some types of flows a simple numerical exit, such as a free-stream exit, can be implemented with only moderate corruption of the solution. However, a general fluid system may have such characteristics as a lack of underlying free-stream flow or nontrivial domain boundaries which prohibit the use of currently documented exit conditions. This work proposes a new boundary condition with the potential for general and accurate simulations of external flows in two or three dimensions.

A third challenge is the proper treatment of body boundaries in Cartesian-grid methods. Because they do not require complex boundary fitted grids to be created, Cartesian-grid solvers have the potential to generate solutions to more complicated problems orders of magnitude faster than conventional fitted-grid solvers. Yet, this speed and generality must not result in a lack of accuracy in the flow around bodies immersed in the fluid domain. A new and general formulation of this problem is presented whereby the analytic form of the fluid equations are altered to ensure exact enforcement of the boundary conditions.

This formulation produces equations which are a superset of those solved in boundary-fitted methods and

could be used to derive adjustments to the discrete algebraic equations as in Cut-Cell methods. Instead of altering the discrete form of the equations we present a simple implementation akin to Immersed-Boundary methods which automatically maintains the solver’s order of accuracy. While any type of body geometry representation could be incorporated into this method, Non-Uniform-Rational-Basis-Splines (NURBS) are used in this work which allow for increased efficiency and flexibility in immersing the body on the grid.

In the fourth section of this paper an assessment is made of the ability of the current Cartesian-grid method to use “slender ship”, or 2D+t, assumptions to preform reduced simulations of bow wave flows. Completely resolving naval hydrodynamic flows using brute force techniques is impossible due to the required computing expense. 2D+t models offer a simple alternative which reduces that cost by multiple orders of magnitude. Within this framework, experimental wave maker results are used to validate the code. Limitations due to this method as well as practical tangential boundary conditions for Cartesian-grid methods are discussed and one viable solution is presented.

When proposing these modeling advances it is important to retain the inherent advantages of Cartesian-grid methods. The proposed solutions for (i) free surface advection, (ii) exit boundary conditions, and (iii) solid-body treatment are as simple and general as possible. These capabilities are developed with general large-scale simulations in mind and have benefits which allow comprehensive simulation of ship hydrodynamic flows on Cartesian grids.

FREE SURFACE ADVECTION METHOD

Volume-of-Fluid (VOF) methods allow topologically complex surfaces to be treated generally by tracking the volume of each fluid instead of the interface location. Current methods to propagate the fluids allow volume to be lost at the free surface leading to “stepping” of the interface and spurious flotsam and jetsam. In this section a simple set of requirements are detailed to avoid this problem, and a general advection algorithm is proposed which exactly conserves the total fluid volume and maintains a sharp air-water interface.

VOF techniques use a field f to specify the fraction of each finite volume which is filled with one type of fluid in order to track the free surface. By definition, the fraction

f must satisfy the condition

$$0 \leq f \leq 1 \quad (6)$$

at all times. This condition is difficult to maintain and standard advection methods result in cells which are less than empty and more than full. Volume must be added or removed *ad-hoc* from these cells introducing significant inaccuracies into the simulations at the free surface.

The volume fraction of fluid in an incompressible flow is advected by solving the conservative equation

$$\frac{\partial f}{\partial t} + \vec{\nabla} \cdot (f\vec{u}) = 0. \quad (7)$$

Pilliod and Puckett (2004) present one of many suggested unsplit advection algorithms to solve (7) for two-dimensional flows based on geometric flux integrations along approximated characteristics of the velocity field \vec{u} . These methods do not typically conserve volume and are very complicated, particularly for three-dimensional flows.

The standard simplification is to use an operator-split advection method. In an operator-split algorithm, the non-conservative advection equations

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x_d} (fu_d) = f \frac{\partial u_d}{\partial x_d} \quad \text{for } d = 1 \dots \mathcal{N} \quad (8)$$

are solved sequentially for progressive updates of f for each of the \mathcal{N} spatial dimensions. By treating only one velocity component at a time, the calculation of the advected volume fraction through each face can be analytically determined with relative ease, even for planar surface reconstructions in three dimensions (Scardovelli and Zaleski 2000). A typical two-dimensional reconstruction is shown in Figure 1. In this figure the velocities are scaled by $\Delta t/\Delta x$ making them local Courant numbers. With this scaling a donating region upwind of each face can be defined with width equal to the velocity magnitude. The dark fluid within each region will be fluxed into the next cell. In Figure 1, the fluid in the bottom-right corner of cell (i, j) is in two donating regions. To avoid the possibility of fluxing the same fluid into two different cells the surface must be reconstructed after each sweep of an operator-split method.

The divergence term is not canceled from (8) because at each step of the algorithm the volume fraction is advected in a one-dimensional flow *which is not divergence free*. Without the velocity stretching term there is no way to

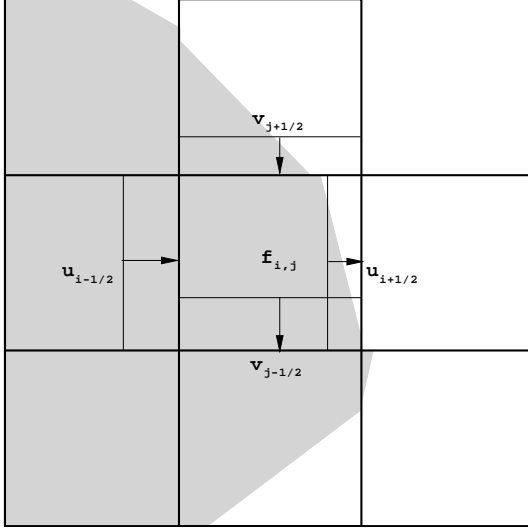


Figure 1: 2D diagram of a typical linear VOF surface reconstruction on a 3x3 block of cells with scaled velocity components. Using the standard operator-split treatment, the donating regions are right-cylinders upwind of each face.

ensure that (6) is met after each step of the algorithm. In Figure 1 more fluid is being fluxed in from cell $(i - 1, j)$ than space left after fluxing out to cell $(i + 1, j)$, meaning that cell (i, j) would overflow in the first sweep. Scardovelli and Zaleski (2003) investigate this effect in detail geometrically and develop a fairly simple two-dimensional operator-split Eulerian implicit Lagrangian explicit (EI-LE) method which conserves the global volume fraction exactly. In that method, (8) is integrated implicitly with for $d = 1$ and explicitly with Lagrangian flux treatment for $d = 2$. Using the same velocity scaling as above, the discrete form of the implicit and explicit integrations are given by

$$f_{i,j}^{n+1/2} = \frac{f_{i,j}^n - (F_{i+1/2}^n - F_{i-1/2}^n)}{[1 - u_{i+1/2}^n - u_{i-1/2}^n]} \quad (9)$$

$$f_{i,j}^{n+1} = f_{i,j}^{n+1/2} \left[1 + v_{j+1/2}^n - v_{j-1/2}^n \right] - (G_{j+1/2}^{n+1/2} - G_{j-1/2}^{n+1/2}) \quad (10)$$

where F and G are the fluxed dark fluid in the first and second directions. Puckett et al. (1997) propose the same

splitting without the Lagrangian flux calculation. Rider and Kothe (1998) propose integrating first explicitly, then implicitly.

Aulisa et al. (2003) demonstrate by two-dimensional algebraic mapping that the EI-LE scheme is the only one of the three that is volume conserving. It also shows that this is dependant upon a strictly two-dimensional divergence free velocity field. The mathematical analysis in that work is straightforward but does not help suggest a general solution for three-dimensional flows.

Breaking the problem of conservation into a short list of requirements clarifies the issues. Given that

1. The flux terms are conservative, *and*
2. The divergence term sums to zero *and*
3. No clipping or filling of a cell is needed due to violation of (6) at any stage

then the algorithm *must* conserve f to machine precision. The first requirement ensures that any fluid going into one cell is coming out of another and the second ensures that there is no net source term added to the advection equation. Along with the third requirement, it is then guaranteed that there is no net change in f in the fluid domain regardless of the dimensionality of the system. By multiplying the divergence term by f^n in the explicit step and then f^{n+1} in the implicit step, Rider and Kothe (1998) create a net source term, violating requirement 2. Implicit integration violates requirement 1 by scaling the fluxes by a local stretching term, bracketed in (9). This scaling term is not consistent on both sides of the cell face, meaning the effective flux is not conservative. The EI-LE scheme maintains global conservation despite this by using the Lagrangian flux calculations and relying on the bracketed term in (9) to be exactly canceled by the bracketed term in (10) due to two-dimensional continuity. This cancellation is not extendable to three dimensions.

However, these requirements are not insurmountable and a simple operator-split algorithm which meets them is

$$f^a = f^b - \Delta_d F_d^b + g^n \Delta_d u_d^n \quad \text{for } d = 1 \dots \mathcal{N} \quad (11)$$

where Δ_d denotes differences in the d direction and g is a function to be determined. The b and a superscripts indicate the *current* and *updated* value of f respectively. As in equations (9) and (10), the fluxes are always calculated using the current value of f to avoid fluxing the

same fluid twice. The key features of this scheme are the use of conservative fluxes and the constant g^n field multiplying the divergence term. Thus requirements 1 and 2 are met, and it only remains to examine requirement 3.

The form of (11) emphasizes that the divergence correction term is offsetting the dark fluid flux with an approximate flux proportional to the stretching. A definition of g is needed which maintains (6) for any configuration, and we propose the simple form

$$g^n \equiv \begin{cases} 1 & \text{if } f^n > 1/2 \\ 0 & \text{else} \end{cases} \quad (12)$$

making g a sharp version of f . It might seem intuitive to let $g^n = f^n$, but setting the approximate flux proportional to the fraction in the central cell does not guarantee conservation. Scardovelli and Zaleski (2003) give a detailed example of this for essentially the situation in Figure 1. Using $g^n = f^n$ for this case results in overfilling in the first step because the dilatation is small and the influx is large. Using the definition of g above enhances the influence of the stretching term and with the restriction

$$|u_d|, |\Delta_d u_d| < \frac{1}{2\mathcal{N} - 2} \quad \text{for all } d \quad (13)$$

guarantees that (6) is met at all stages of the algorithm. The velocity restriction limits the acceptable time step; yet our experience has shown that it is conservative, particularly for three-dimensional flows.

Even with the restriction on the time step, all three requirements are met and the algorithm will conserve volume fraction to machine precision. A series of canonical three-dimensional simulations were run to verify this capability and quantify errors in other methods. The test case is a high amplitude standing air-water wave and the Cartesian-grid solver described in the previous section was used to run the tests. The computational domain is a cube, with $\Delta x/L = 0.025$, $\Delta t/T = 0.005$ and reflection boundary conditions set on all walls. A sinusoidal free surface with $A = 0.3L$, $\lambda = 2L$ and an average elevation of $z = 0.5L$ is used as the initial condition. The slope is well above the Stokes limit and results in a highly nonlinear fully three-dimensional wave with overturning as seen in Figure 2.

Five advection methods were tested, the first being a baseline test with no stretching term included. The second two are extensions of methods such as that of Rider and Kothe (1998) to three-dimensional flows. The first (E-E-E) uses all explicit integrations of the form of (10) and the

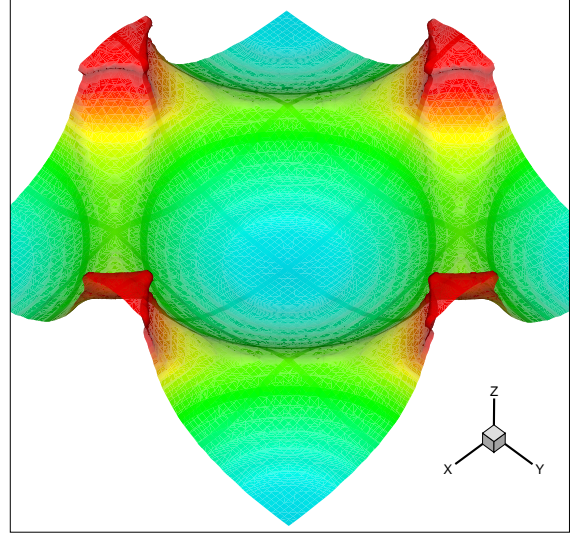


Figure 2: Visualization of the free surface for the 3D wave test at $t = 0.5$. Contours indicate elevation. The nonlinear wave has nearly impinged on the top of the computational domain in this image.

second (I-I-E) uses implicit integration of the form of (9) for the first two steps. Next is the current method using g as defined in (12) and then using $g^n = f^n$ for comparison. The results are shown in Table 1. The first two columns give the L_1 and L_∞ norms of the global volume loss in the domain after each use of the advection method. The third column gives the percentage change in total water volume in the domain after one wave period. The baseline gives by far the worst performance, with a net mass loss of 12% after only one wave period. The second and third rows show around an order of magnitude decrease in error compared to neglecting the stretching term altogether but still leave room for improvement.

The current method demonstrates nearly exact conservation of mass after each step and overall. The average values of $|u|_{max}$ and $|\Delta u|_{max}$ at the interface are around $1/3$ in this test. Despite this violation of (13), the volume loss is negligible. The final row shows the results when setting $g^n = f^n$. While the volume of water is not conserved exactly, there is still an order of magnitude decrease in error as compared to the second and third rows.

Algorithm	L_1 norm	L_∞ norm	% Change
Baseline	2.53e-3	1.40e-2	-12.63%
E-E-E	3.03e-4	2.51e-3	-1.42%
I-I-E	1.22e-4	1.34e-3	+0.81%
Current	1.34e-12	6.83e-12	-0.00%
$g^n = f^n$	9.69e-6	1.51e-4	+0.04%

Table 1: Global mass loss measurements for a high slope 3D standing wave test case. The first two columns refer to the volume loss at each step, and the final column gives the total change after one wave period.

EXIT CONDITION

Exit conditions are required for external incompressible flow simulations and are notoriously difficult to enforce accurately. In practice poor exit conditions require large computational domains and short simulation times to avoid corrupting the solution. This reduces the computational efficiency and reliability of the simulation. In this section a new exit boundary condition is presented which minimizes reflections for general fluid systems.

Unlike the free surface or a no-slip wall, an “exit” is a purely numerical boundary. The ideally semi-infinite domain of an external flow is truncated by the exit boundary to make the solution trackable. In establishing an exit boundary, we are inherently assuming that the flow phenomena of interest are within the numerical domain and independent of the flow beyond the exit. Exit conditions must be set despite this assumption of independence because the pressure Poisson equation is elliptic and requires information on every boundary to ensure a globally divergence-free velocity field.

One method of modeling an exit is to construct a numerical beach to absorb the energy out of exiting waves. Such methods are often ineffective and require large buffer regions, lowering computational efficiency. In Dommermuth et al. (2004) a body force method is developed to drive the flow at the exit to match the free stream conditions, and Ferziger and Peric (2002) suggest using 1D extrapolation to set the velocity profile at the exit. While these methods can be applied in limited cases they rely on a strong underlying convective flow to wash away the errors resulting from their modeling inaccuracies.

A more advanced unsteady condition is a wave exit, such as developed by Orlanski (1976). The typical

Orlanski-type boundary condition applied to the velocity vector is

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)\vec{u} = 0 \quad (14)$$

where c is the, as yet undetermined, wave speed and the x -coordinate represents the direction normal to the exit. When using this condition, the mass flux through the boundary is prescribed and a compatible Neumann condition must be set on the pressure. Reasonable choices of the wave speed c are the free-stream velocity or some other natural velocity metric, such as \sqrt{gL} . In Hurt (1999), a field of exit speeds are determined by solving (14) for c one point interior to the exit. However, this method is unstable if the momentum equations are integrated explicitly.

Regardless of how the wave speed is chosen, if there are multiple waves with differing speeds exiting the domain, (14) will produce reflections for waves with speeds not equal to c . The Higdon condition (Higdon 1994) has been developed to circumvent this difficulty. The condition is a wave product

$$H_J : \prod_{j=1}^J \left(\frac{\partial}{\partial n} + \frac{1}{c_j}\frac{\partial}{\partial t}\right)\vec{u} = 0 \quad (15)$$

where c_j are the set of exiting wave speeds. Increasing the number of wave speeds reduces the energy reflected back into the domain. Derivation and implementation of this conditions for the two-dimensional wave equation is given by Givoli et al. (2003), but applying this boundary condition to incompressible free-surface flows is nontrivial.

Additionally, the requirement of maintaining the divergence-free condition on the velocity means that neither (14) nor (15) can be applied to the exiting component of the velocity in their current form. This is because a velocity boundary condition that determines the mass flux through the exit must be globally compatible with the other boundaries. There is no value of c that will ensure this global conservation, and so regardless of the choice of wave speed (or speeds) the calculated velocities must be adjusted. The standard solution is to integrate the mass fluxes on the other domain boundaries and make an *ad-hoc* adjustment to the exit velocity field such that global conservation is maintained. This is only possible if the mass fluxes on all other boundaries can be readily determined *a priori*, which is not true for general cases.

Formally, this adjustment to the x -component of the velocity vector can be written as

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)u = -\alpha \quad (16)$$

where α is some, hopefully small, deviation from a true exiting wave form. The terms above are readily correlated to those of the x -component of the momentum equation

$$\left(\frac{\partial}{\partial t} + \vec{u} \cdot \vec{\nabla}\right)u = -\frac{1}{\rho}\frac{\partial p}{\partial x}. \quad (17)$$

The time derivatives match and $c\frac{\partial u}{\partial x}$ is a one-dimensional model of the convective acceleration. The remaining term is the pressure force, which leads us to formulate a boundary condition of the form

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)u = -\frac{1}{\rho}\frac{\partial p}{\partial x} \quad (18)$$

for the exiting component of velocity. Adjusting the standard Orlanski exit in this way allows the pressure field to fulfill its normal role of projecting the velocity onto a solenoidal field. Therefore, no global flux integrations are required and the system will conserve mass locally and globally to the tolerance of the pressure solver.

When using (18) a Dirichlet boundary condition must be set on the pressure equation. Hydrostatic pressure could be set at the exit, but a wave condition for the pressure

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)p = 0 \quad (19)$$

is the most consistent option. The boundary condition is applied at the exit using linear interpolation

$$\bar{P}_e = \frac{1}{2}(P_E + P_I) \quad (20)$$

where \bar{P}_e is the interpolated exit value and P_E and P_I are the cell-centered values to the exterior and interior of the exit respectively. Second-order central differences and Runge-Kutta integration in time then give the boundary equations as

$$\bar{P}_e^{n+1/2} = \bar{P}_e^n - \frac{c\Delta t}{\Delta x}(P_E^n - P_I^n) \quad (21)$$

in the predictor step and

$$\begin{aligned} \bar{P}_e^{n+1} = & \frac{1}{2} \left[\bar{P}_e^n + \bar{P}_e^{n+1/2} \right. \\ & \left. - \frac{c\Delta t}{\Delta x} \left(P_E^{n+1/2} - P_I^{n+1/2} \right) \right] \end{aligned} \quad (22)$$

in the corrector step. These equations are substituted into the pressure equation to eliminate references to the exterior pressure value. While modifying the pressure equation at the exit in this manner does entail some additional complexity it is the only method presented which can be used to model any general free-surface flow.

To quantify the performance of these exit condition a series of tests are run using two-dimensional Cauchy-Poisson waves. The domain has an aspect ratio of 2 and is discretized using $\Delta x/L = 0.04$ and $\Delta t/T = 0.02$. The exit conditions are set on the far right boundary at $x = 4L$ and free-slip conditions are set on all other boundaries. The initial free-surface elevation is set to $\eta(x)/L = 1 + \frac{1}{2}\exp(-\pi x/L)$. To measure the error induced by the exit, a reference solution is generated by doubling the length of the numerical domain and using the standard Orlanski exit with $c = 1.5L/T$ at $x = 8L$. Figure 3 shows the initial condition from $x = 0.4L$ along with subsequent snapshots in a waterfall plot for this reference solution. In this figure, time progresses by moving down the page, and the waves travel from left to right. Four nonlinear waves of decreasing amplitude, wavelength and speed can be easily identified in the figure. The figure also shows that there is little reflection into the domain during this time period justifying our use of this solution to estimate errors in the simulations on the truncated domain.

Four types of exiting conditions are tested, the first of which is a baseline case using a free-slip wall condition. This condition will give pure reflection. The second exit condition uses linear extrapolation to determine the velocity profile at the exit. The u component was corrected in by global flux integration to ensure mass conservation. The third condition is the standard single-speed Orlanski exit described by (14) using a global integration adjustment to determine α . The last condition is the pressure adjusted wave exit using (18) and (19). The wave speed is set to $c = 1.5L/T$ for all the wave cases. Figure 4 shows the results from all of these tests along with the reference solution as the second wave reaches $x = 4L$ at $t = 3.36T$. Contours of $\frac{\partial p}{\partial x}$ are plotted within the water to aid in comparing the methods. By this time, the large first wave has reflected back across the domain in Figure 4(b) and the comparison to the reference solution is already poor. The extrapolation exit condition is observed to give essentially the same solution as the reflection exit. The two wave exits compare much better to the reference solution.

To establish a quantitative comparison, the L_2 norm

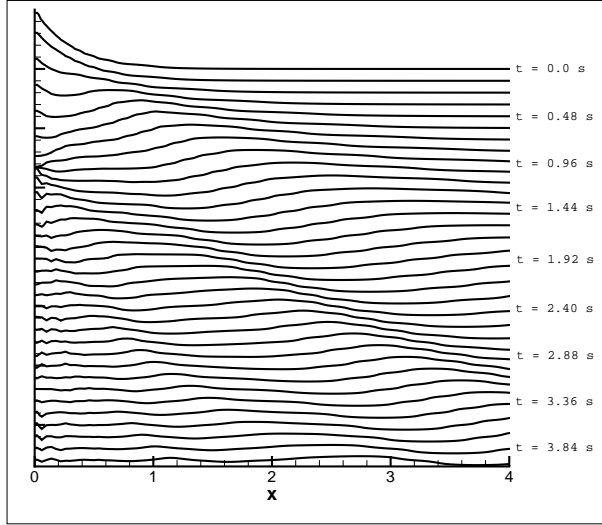
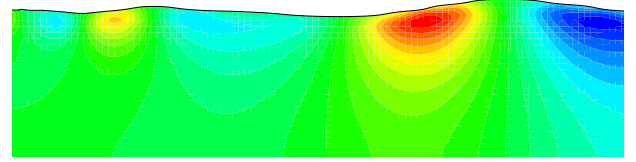


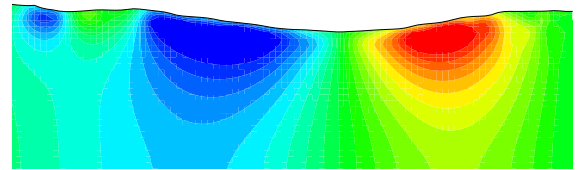
Figure 3: Free surface elevation waterfall plot of the reference solution. Each line is $6\Delta t = 0.12T$ later than the line above it. The view is truncated at the $x = 4L$ where the tested numerical exit conditions were enforced.

of the error in the free surface elevation in the truncated domain is calculated as a function of time. The results are shown in Figure 5 for each exit condition and for a range of c values. The error in the pure reflection condition is seen to fluctuate in time as the waves reflect from one end of the truncated domain to the other. The Euler equations generate no physical dissipation to damp out the wave energy yet the magnitude of error remains well bounded in time. This is in contrast to the extrapolation exit which has similar characteristics to the reflection boundary initially but rapidly becomes unstable and diverges completely at $t = 9T$. This instability in the extrapolation exit demonstrates that spurious information generated at the exit has fully corrupted the solution.

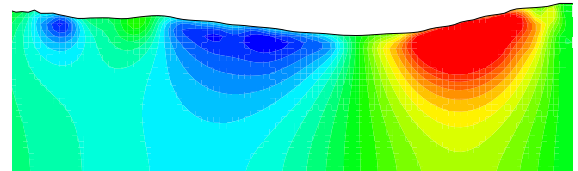
The next six lines of Figure 5 show the wave exit with global-integration correction (labeled Wave 1) and pressure correction (labeled Wave2) using wave speeds of $c = 0.5, 1.0$ and $1.5L/T$. All six wave exits show a significant decrease in error and a fast overall trend towards zero error in time. By $t = 10T$ nearly all waves have traveled out of the truncated domain, and the remaining error is likely indicative of reflections in the reference solution. For all choices of wave speed, we see that the pressure corrected exit has comparable errors to the exit corrected by global-integration. As (18) is more



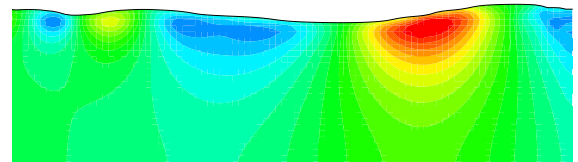
(a) Reference solution



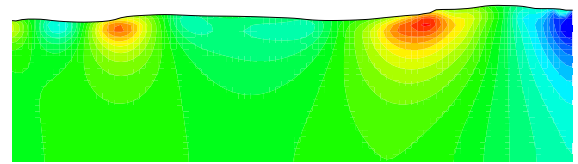
(b) Reflection condition



(c) Extrapolation exit



(d) Wave exit with global-integration adjustment



(e) Wave exit with pressure adjustment

Figure 4: Contours of $\frac{\partial p}{\partial x}$ in the water for the exit conditions at $t = 3.36T$. A wave speed of $c = 1.5T/L$ was used for all wave cases.

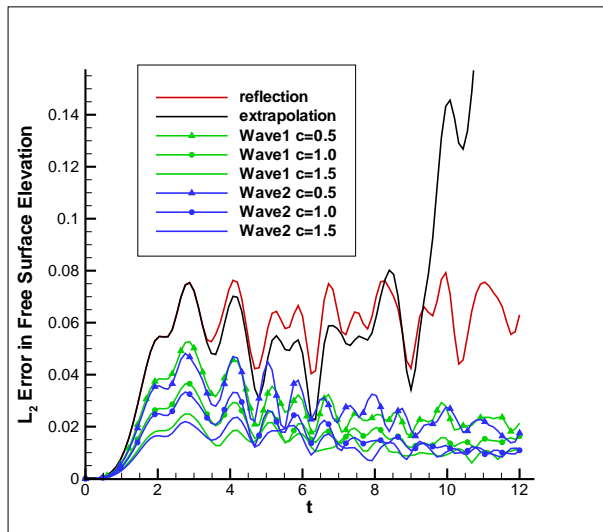


Figure 5: Measurements of the L_2 spatial norm of free surface error as a function of time. The wave exit with global-integration correction are labeled Wave 1, and the pressure corrected exits are labeled Wave 2.

physically relevant than (16) and allows comprehensive treatment of free-surface flows, we conclude it to be a significant advancement.

However, the measurements shown in figure 5 still show reflections which could corrupt solutions. The Higdon condition may be used to extend the method to multiple wave speeds but solution algorithms for such methods are complicated, particularly for variable density flows. Simplifying these numerical methods is a topic of current research.

BODY BOUNDARIES

Treatment of body boundaries are of particular concern in Cartesian-grid methods. The methods currently available have limitations in applications or accuracy or have complicated implementations. A general framework is presented in the next two sections to formulate equations of motion which accurately simulate complex flows around solid geometries. A simple implementation is proposed which automatically maintains the order-of-accuracy of the general flow solver. While any surface representation could be used with this method, we present a representation which increases ease of use and efficiency.

Thus far, two primary methods exist in the literature to enforce the effects of solid bodies in Cartesian-grid simulations: Immersed-Boundary methods and Cut-Cell methods. The Immersed-Boundary method was developed for use in fluid-structure interaction problems, specifically biological fluid dynamics. Peskin (2002) gives a detailed review. In general, the elastic body equations are solved on an explicitly defined surface mesh, while the fluid equations are solved on a Cartesian grid. The two simulations are linked by applying the reaction force of the body on the fluid and advecting the body in the resulting flow. The localized forces and body velocities are calculated using a smoothed approximation of the Dirac delta function. Because the body is advected by the flow the interactions are restricted to flexible bodies and systems that are not mathematically stiff.

The so called Cut-Cell methods alter the discrete form of the fluid equations of motion near the body to account for its presence. There are a great variety of these methods, and the body may either be explicitly defined by a mesh or implicitly defined by a field (Udaykumar et al. 2001). The changes to the discrete equations often involve interpolating boundary values such as in Chimera methods and altering the local grid metrics such as with non-orthogonal grids. Other alterations have also been researched such as locally changing the grid from staggered to collocated (Gilmanov and Sotiropoulos 2005) and setting up extrapolated “ghost-cells” (Tseng and Ferziger 2003) within the domain. The difficulties with these methods are their complexity, computational expense and maintaining the flow solver’s order of accuracy and stability (Ye et al. 1999).

Dommermuth et al. (1998) avoids these issues in developing a simple body force method suitable for rigid no-slip bodies such as ships. A body force term is added to the discrete momentum equations within the body proportional to the velocity error. This force drives the velocity of the fluid within the body to the prescribed body velocity exponentially in time. As such, the boundary condition is treated as a goal-state rather than an instantaneous restriction on admissible velocity fields.

Expanding upon that work, a new method is presented to constrain the flow to the known instantaneous velocity conditions on a general, dynamic body \mathcal{B} for two and three-dimensional flows. A distance based normalized

delta function, or switch function, defined as

$$\delta'(\vec{x}) \equiv \begin{cases} 1 & \text{for all } \vec{x} \in \mathcal{B} \\ 0 & \text{else} \end{cases} \quad (23)$$

is used to alter the analytic form of the governing equations. This is in contrast with Cut-Cell methods which change the discrete form of the equations near the body. By directly imposing constraints on the velocity field the body will drive the flow instead of being advected by it, allowing for the treatment of high-stiffness bodies such as ship hulls.

No-Slip Condition

In this subsection equations to enforce the no-slip body boundary condition on a general topological body \mathcal{B} immersed in the fluid domain \mathcal{R} are derived. We are given the velocity vector \vec{U} of \mathcal{B} as well as the distance from any fluid point in the domain to the body surface. The no-slip condition is then stated as

$$\vec{u} = \vec{U} \quad \text{for all } \vec{x} \in \mathcal{B}. \quad (24)$$

To derive a no-slip equality for use in the momentum equation, (24) is substituted into (2) to give

$$\frac{\partial \vec{U}}{\partial t} + \frac{1}{\rho} \vec{\nabla} p - \vec{r} = 0 \quad \text{for all } \vec{x} \in \mathcal{B}, \quad (25)$$

which is commonly used as a starting point for deriving solid-body boundary conditions for the pressure. In this method we instead multiply (25) by the function δ' . Its definition ensures that this product is zero throughout the domain and therefore may be added to (2) resulting in the altered momentum equation

$$\frac{\partial \vec{u}}{\partial t} = (1 - \delta') \left(\vec{r} - \frac{1}{\rho} \vec{\nabla} p \right) + \delta' \frac{\partial \vec{U}}{\partial t}. \quad (26)$$

Taking the divergence gives the corresponding pressure equation as

$$\begin{aligned} \vec{\nabla} \cdot \left(\frac{1 - \delta'}{\rho} \vec{\nabla} p \right) = \\ \vec{\nabla} \cdot \left((1 - \delta') \vec{r} + \delta' \frac{\partial \vec{U}}{\partial t} - \frac{\partial \vec{u}}{\partial t} \right). \end{aligned} \quad (27)$$

Equations (26) and (27) are identical to equations (2) and (3) except on \mathcal{B} where the switch function is

used to transition the equations to an enforcement of the no-slip condition. Technically speaking, the above equations enforce the time derivative of (24), but when the momentum equation is integrated in time, the exact boundary condition is recovered on \mathcal{B} . This analytic transition from the field equations to the boundary conditions differs significantly from the standard body force or Immersed-Boundary formulations.

When these equations are applied to the boundaries of the numerical domain, they produce exactly the same discrete formula obtained by applying boundary conditions to a boundary fitted grid. This demonstrates that equations (26) and (27) are generalizations of the equations used in fitted grid methods, allowing the body to lay anywhere in the domain instead of restricting it to coincide with the domain boundary. Because of this generality the above equations can be formulated for any number of arbitrary bodies with any given velocity. Also, as \mathcal{B} need not be a material surface, the inflow and outflow boundaries can be modeled with this formulation by setting \vec{U} appropriately.

No-Penetration Condition

The method is next extended to the no-penetration condition, given by

$$\vec{u} \cdot \vec{n} = \vec{U} \cdot \vec{n} \quad \text{for all } \vec{x} \in \mathcal{B} \quad (28)$$

where \vec{n} is the unit normal vector of the surface of \mathcal{B} .

This boundary condition is substituted into the normal projection of (2) to give

$$\begin{aligned} \left[\vec{n} \cdot \left(\frac{\partial \vec{U}}{\partial t} + \frac{1}{\rho} \vec{\nabla} p - \vec{r} \right) + \frac{\partial \vec{n}}{\partial t} \cdot (\vec{U} - \vec{u}) \right] \vec{n} = 0 \\ \text{for all } \vec{x} \in \mathcal{B} \end{aligned} \quad (29)$$

where the second term within the brackets is arbitrary and can be set to zero. To see this, note that the length of \vec{n} is constant, meaning its time derivative must lie in the tangent plane to the body surface. As the tangential velocity of \vec{U} is arbitrary it can always be set so that the inner product vanishes.

Using the same procedure as in the previous section, (29) is multiplied by δ' and added to the governing equations. The resulting no-penetration momentum and pressure equations are

$$\frac{\partial \vec{u}}{\partial t} = (\mathbf{I} - \delta' \mathbf{N}) \left(\vec{r} - \frac{1}{\rho} \vec{\nabla} p \right) + \delta' \mathbf{N} \frac{\partial \vec{U}}{\partial t} \quad (30)$$

and

$$\begin{aligned} \vec{\nabla} \cdot \left((\mathbf{I} - \delta' \mathbf{N}) \frac{1}{\rho} \vec{\nabla} p \right) = \\ \vec{\nabla} \cdot \left((\mathbf{I} - \delta' \mathbf{N}) \vec{r} + \delta' \mathbf{N} \frac{\partial \vec{U}}{\partial t} - \frac{\partial \vec{u}}{\partial t} \right), \end{aligned} \quad (31)$$

where \mathbf{I} is the identity tensor and $\mathbf{N} \equiv \vec{n}\vec{n}$ is the normal dyad. Although similar in form to the no-slip equations, the no-penetration equations feature tensor products instead of simple multiplication. This additional complexity arises because (28) is a scalar equality and can therefore only make a rank-one adjustment to the governing equations. The adjustment is made through the rank-one dyad \mathbf{N} , leaving the $(\mathcal{N} - 1)$ tangential components of the momentum equation unchanged by the presence of the body. These tangential equations are then free to carry any user defined slip model including the no-slip condition. This topic is explored further in the following sections.

The tensor products make numerical solution of the no-penetration equations more laborious. The source term of (31) is easily constructed, but the Poisson coefficients give rise to cross derivatives. These cross derivatives are needed to cancel out the normal component of the pressure gradient. As they arise in this general formulation they must be treated regardless of the implementation being Cut-Cell, fitted-grid, or the method described in the next section. In this work, the cross derivatives have been treated with differenced-correction whereby old values of the cross derivatives are used and then corrected after the pressure field is calculated. This maintains the simplicity of the pressure solver and allows general treatment of the equations throughout the domain.

Numerical Implementation

In a flow solver the momentum equation is evaluated at discrete points in the domain which, in general, do not exactly coincide with the surface \mathcal{B} , allowing the surface to “hide” from the flow. There are many ways to overcome this issue, the best established of which is to use a non-orthogonal boundary-fitted grid. Another option is to alter the discretization of the momentum and pressure equations near the body, as in Cut-Cell methods. An example is the slip condition presented in Dommermuth et al. (2005), which is derivable by integrating (31) over each finite volume.

In this work a smoothed approximation of the analytic delta will be used instead of altering the discretization scheme. Similar to the form used by the Immersed-Boundary method, we define the smoothed switch function as

$$\delta'(\vec{x}) = \begin{cases} \frac{1}{2} (1 + \cos(d \frac{\pi}{\epsilon})) & \text{for all } |d| < \epsilon \\ 0 & \text{else} \end{cases}, \quad (32)$$

where d is the distance from the point \vec{x} to the surface, and ϵ is the width of the numerical delta function. This definition is used to allow modeling of thin sheets in our solver. An alternate definition more akin to the smoothed heavy side function would be appropriate for thick solid bodies. Because of the use of (32), discontinuities resulting from the presence of the body will be smoothed over the width ϵ . The current method is therefore not a “sharp-interface” method, and care must be taken in the setting of ϵ such that accuracy is not lost. The smoothing width was set to $\epsilon = 2\Delta x \sqrt{\mathcal{N}}$ for all tests in this work and is demonstrated below not to reduce solution accuracy. The benefit of this method is its trivial and general implementation. Also, because the method does not alter the calculation of derivatives the order-of-accuracy of the bulk flow solver is automatically maintained.

Two simple tests are presented to demonstrate the ability of this boundary condition enforcement technique to accurately simulate unsteady free-surface flows. In both, a two-dimensional tank with aspect ratio 2 is simulated using $\Delta x = 0.0125L$ and $\Delta t = 0.00267T$. The first case is a high amplitude standing wave test with $A = 0.2L$ and $\lambda = L$. An image of the resulting nonlinear wave at time $t = T$ is shown in Figure 6(a). Note that the simulation is symmetric on either side of the $x = 0.5L$. To test the proposed method a vertical wall is placed at that location and a constant free surface height of $0.05L$ is set on the right side while the wave initial condition is maintained on the left. This simulation will thus test the method’s ability to enforce the no-penetration condition and quantify the errors due to smoothing the boundary. The result using the current method is shown in Figure 6(b). For comparison the result for the same simulation using the body force method is shown in Figure 6(c). Unlike the body force method, no fluid is transmitted through the wall, and errors due to smoothing are very small.

In the second case a tank half filled with water is

quickly displaced to the right by $0.2L$ and then held steady. This can be modeled using a frame of reference that moves with the tank, accounting for the acceleration by the application of a uniform body force. It can also be simulated by enforcing the no-penetration condition on the moving vertical walls of the tank using a stationary frame of reference. Therefore, the same result should be generated using a fitted grid or moving immersed boundaries, allowing for a direct comparison. Figure 7(a) and 7(b) show the result for the fitted and immersed grid respectively at $t = 2.4T$. The images show that the comparison between the methods is excellent, even for this highly complex flow with dynamic boundaries.

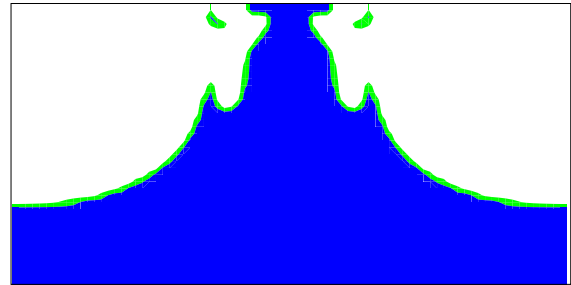
Advanced Body Representation

All flows with bodies must describe the geometry in some way and the most useful description is problem dependant. When solving biological fluid-dynamics problems with the Immersed-Boundary method, a simple algebraic mesh representation is logical because the elastic body equations need to be solved on a mesh. However, when solving flows with the method described in the previous section a more powerful set of functions can be chosen to describe the geometry.

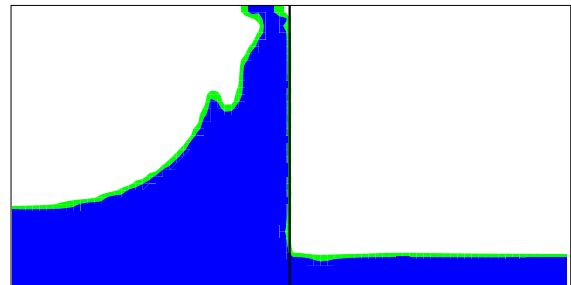
NURBS (Non-Uniform Rational B-Splines) are one of the most popular tools used to represent lines and surfaces in computer aided design and graphics, and are the backbone of such programs as Rhino and FastShip. This is because of their efficient implementation, their intuitive control point interface, and the smoothness properties of the resulting forms. In this work, the NURBS description used to design a solid body is maintained in the computational analysis of the flow around that body. This eliminates all need for gridding and assures the designer that WYSIWYG¹. Another consideration is that NURBS surfaces have far fewer parameters than algebraic grids making them better suited for shape optimization problems.

In order to enforce body boundary conditions on a Cartesian grid every point on the grid must know the distance d to the nearest point on the body. Additionally, topological parameters such as the normal \vec{n} are required. Although nowhere near as time consuming as creating a fitted mesh, determining these values can be a slow process. On an algebraic mesh the distance to the body is found by an exhaustive search of every point, line,

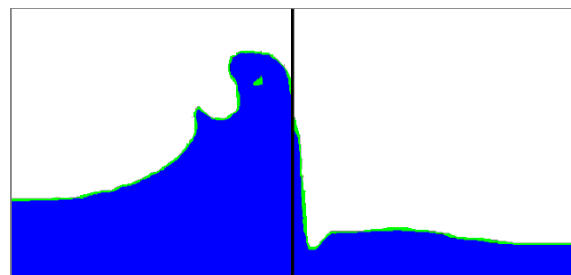
¹What You See Is What You Get



(a) Symmetric standing wave in full tank

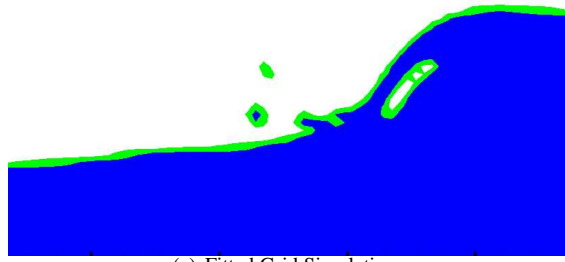


(b) Standing wave in split tank using current method

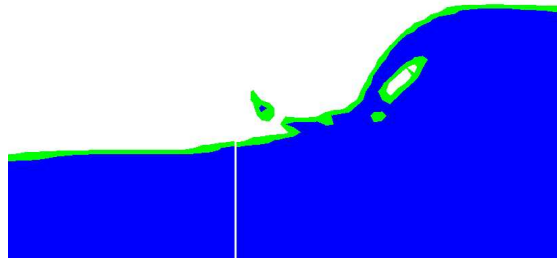


(c) Standing wave in split tank using body force method

Figure 6: Images of high amplitude standing waves in a tank. Cells full of water are colored blue, cells full of air white, and partially filled cells green. Figure (a) shows the baseline case with no immersed surface. The domain has been split in two by a vertical wall in figures (b) and (c), using the current formulation and the body force formulation respectively.



(a) Fitted Grid Simulation



(b) Current Method Simulation

Figure 7: Image of sloshing waves in a tank generated by rapid sideways displacement using the same coloring as in Figure 6. Figure (a) shows the baseline using a fitted-grid formulation and Figure (b) shows the result using the current method.

and planar surface on the mesh. The parametric NURBS representation allows the use of a gradient based search for the distance function that is orders of magnitude faster. Formulating the squared distance from a point \vec{x} in the domain to the point \vec{X} on the body as a function of the surface parameter \vec{s} as

$$\psi^2(\vec{s}) = \left| \vec{x} - \vec{X}(\vec{s}) \right|^2, \quad (33)$$

allows the minimum distance function to be defined by

$$d^2 = \min_{\vec{s}} \psi^2(\vec{s}) \quad (34)$$

which can be solved with the Gauss-Newton method for non-linear least squares which has a second-order convergence rate.

To quantify the speed-up observed by using this method, the signed distance function and normal vector for a spherical test geometry are computed on a series of background grids. This is compared to the average accuracy and speed of calculation of the distance function

$R/\Delta x$	NURBS time	Mesh time	Mesh Error
32	0.301e-1	0.201e-1	0.119e-2
64	0.121e+0	0.660e+0	0.178e-3
128	0.181e+1	0.202e+2	0.453e-4
256	0.127e+2	—	—

Table 2: Distance calculation statistics for immersed sphere using mesh and NURBS based surface representations.

and normal using a structured surface mesh. Table 2 shows these results. The distance function and normal vector were found using a single processor computer, and the times (given in seconds) are only meant for relative comparison. The value $R/\Delta x$ is the sphere radius over the grid spacing and therefore proportional to the number of grid points in one direction. The average error in the parameters using the NURBS solver was less than $1e-7$. The mesh-based method was stopped after 45 minutes on the finest grid and the results are not shown for this case. The table shows that the cost of determining the parameters scales linearly with the number of points when using the Gauss-Newton solver, but at least quadratically when using an exhaustive search on the mesh. This is especially important when the geometry is continually changing as the simulation progresses, such as in hull shape optimization and the flexible wave maker problem presented in the next section.

An additional advantage of this representation is that it can be easily adjusted to compute distance function to the cross-sectional lines of any surface. When the distance function is found using a gradient method a lagrange multiplier can be constructed which constrains the set of admissible points on the surface to a particular cross section with no increase in computing time. To create the same distance function using a meshed geometry would require expensive and complicated preprocessing of the body geometry. Figure 8 shows a containership bow with the cross sectional lines. Though the lines are discontinuous at the bulbous bow the current method can handle this discontinuity with no special treatment.

NAVAL SHIP HYDRODYNAMICS APPLICATION

At this point, simple and general methods for treating the free surface, numerical exit boundary, and body have been developed and tested. In this section we will demonstrate the ability of these advanced Cartesian-grid

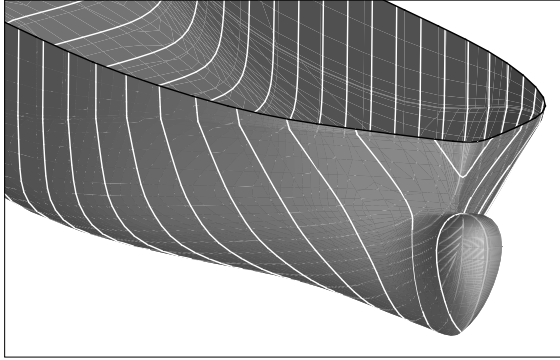


Figure 8: Bow of a general container ship hull-form with bulb. The lines are cross sections of the surface with x -planes, as would be required for 2D+t simulation.

methods to simulate flows with naval applications and use comparisons of those solutions to experimental results to discuss possible reduced models of bow flows and proper tangential boundary conditions for Cartesian-grid methods.

Figure 9 shows the bow wave flow of the model 5415 hull in a 30 knot simulation generated using standard NFA methodologies. This simulation is performed on massively parallel machines using nearly 30 million grid points and grid stretching on the Cartesian mesh. Comparison of these simulations to experimental results for the same test case demonstrate generally good agreement, but the experiments show that the run-up of the bow wave is under-predicted by NFA even for this high resolution run. This error is due to the seven orders of magnitude disparity in relevant length and time scales in naval ship hydrodynamics which can not be resolved directly even with the most advanced brute force methods (Weymouth et al. 2006).

One interesting proposal to deal with the demands of resolving full-scale naval hydrodynamic flows is to adopt the “slender ship” assumptions, modeling the three-dimensional system as a 2D+t flow. 2D+t models simplify the simulation of ship bow waves by assuming that changes in the longitudinal direction are small compared to changes in the transverse directions. Historically, this allowed potential flow simulations of the bow waves around slender vessels to be generated with orders of magnitude savings in computational expense. Fontaine et al. (2000) provide a thorough derivation of the method and present results generated using a nonlinear potential

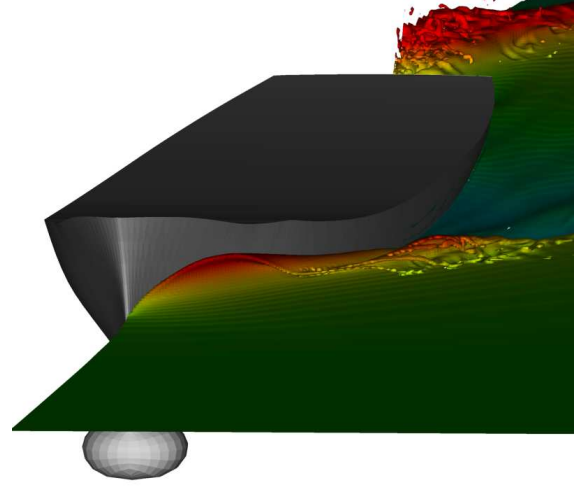


Figure 9: 3D image of the 5415 bow wave flow at 30 knots as simulated by NFA

flow solver.

The pertinent issues may be addressed with a brief introduction to the 2D+t methodology. Using the ship length L and draft D as the relevant dimensional parameters gives the laplace equation for the velocity potential ϕ close to the body as

$$\left(\gamma^2 \frac{\partial^2}{\partial \hat{x}^2} + \frac{\partial^2}{\partial \hat{y}^2} + \frac{\partial^2}{\partial \hat{z}^2} \right) \phi = 0 \quad (35)$$

where $\hat{x} = x/L$, $\hat{y} = y/D$, $\hat{z} = z/D$ and $\gamma = D/L$. If the vessel is very slender then $\gamma \ll 1$ and the equation has no x -dependence to leading order. The kinematic and dynamic free-surface boundary conditions set the requirement that

$$\gamma \frac{U^2}{gL} \gg O(1) \quad (36)$$

and impose a downstream x -dependence on the solutions, but no upstream influence (Fontaine and Cointe 1997). Therefore, the problem is parabolic and may be posed as an unsteady two-dimensional nonlinear system instead of a three dimensional one.

While potential flow solvers are typically used to solve the resulting unsteady two-dimensional flow, the assumptions of potential flow theory prevent it from modeling steep and overturning ship waves such as those shown in Figure 9 for a number of reasons. Firstly, as the run-up is dependant on the near-wall flow it can

not be assumed that the flow is inviscid. Additionally, plunging breakers such as those of Figure 9 are highly rotational. Therefore, a potential function may not be used to describe the velocity field and other means must be used to simulate the flow within this 2D+t framework.

Geometrically, the 2D+t assumptions reduce the three-dimensional fluid problem to a two-dimensional cross section of the flow which moves along the length of the body in time. This models the hull as a deforming curve which can be thought of as a flexible wave maker, pushing out the 2D+t representation of the divergent wave system generated by the body. Shakeri (2005) presents experimental results for a physical realization of this geometric interpretation. In those experiments, a large (3m tall) wave maker was actuated by hydraulics to sweep out the bow of the 2D+t representation of a modified model 5415 hull traveling at 25 knots. The sonar dome was removed from the representation of that hull due to limitations in the wave maker experimental apparatus. This shows that while these experimental results bypass the limitations of potential flow they introduce limitations of their own.

Cartesian-grid methods overcome these shortcomings and those of potential flow and afford the opportunity to study the effect of 2D+t modeling on breaking bow waves generated by realistic ship geometries. As depicted in Figure 8, hulls with bulbs, chines and appendages hulls give rise to discontinuous and multiply connected “wave makers” that only the NURBS body representation and Cartesian-grid methods have the capability of modeling simply and generally. Such hull features will generally violate the slenderness assumptions of the 2D+t framework, and the errors incurred by these effects can be quantified with detailed simulations.

Additionally, Cartesian-grid methods offer the unique capability to quantitatively determine the bounds on the validity of 2D+t models for complex bow flows. 2D+t methodologies demand that a hull with length L and speed U will produce the same wave as a stretched geometry with length $10L$ and speed $10U$. A series of three-dimensional simulations can be run varying the length scale and Froude number of the vessel to determine when stream-wise variations become important. Because the methods proposed in this paper are general enough to model free surfaces, exits and bodies in two or three dimensions, comparisons of simulations with different numbers of dimensions can be made with confidence.

As a first step the data provided by Shakeri (2005) is used to validate the Cartesian-grid capabilities described

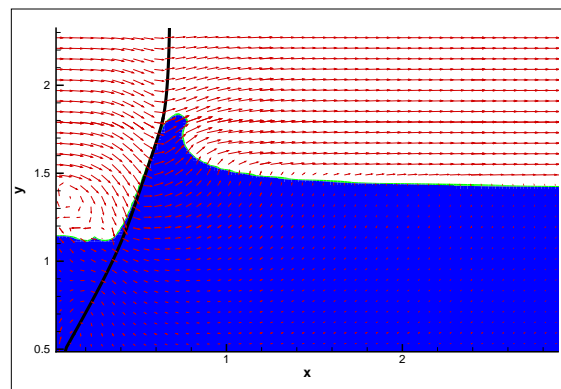


Figure 10: Image of the flexible wave maker and calculated flow on the finest grid. The dark solid line is the wave maker surface, the arrows are velocity vectors and the blue coloring denotes water. Every third velocity vector is shown for clarity. The no-penetration condition is enforced on the body

in the previous section. The wave maker is simulated with the body treatment of the previous section and the position of the wave maker in time is set to exactly duplicate the experimental conditions. A computational domain of $3m$ by $6m$ is defined for the simulations and a series of time steps and grids spacings are used to judge their influence on accuracy. Coarse, medium and fine background grids with spacings of $\Delta x = 0.04m, 0.028m$ and $0.02m$ respectively are used. The pressure corrected wave exit developed above is used with $c = 1 \frac{m}{s}$. Figure 10 shows a snap-shot of the wave maker and simulated flow using the no-penetration condition on the finest grid level. As can be seen from the velocity vectors in that image, the wave maker is sweeping from left to right with the lower end pinned. In this simulation, fluid was allowed to flow in behind the wave maker using a one-way periodic condition, ensuring that the continuity condition is met in that region. The flow behind the wave maker has no influence on the flow exterior to the wave maker. The image demonstrates that the fluid has been allowed to run smoothly up the side of the body boundary and that the no-penetration condition has been exactly enforced in the normal direction.

Figure 11 shows a multiple exposure image of the wave maker position and free surface elevation at sequential times in the simulation. The wave resulting from this motion is highly energetic and goes through a series of

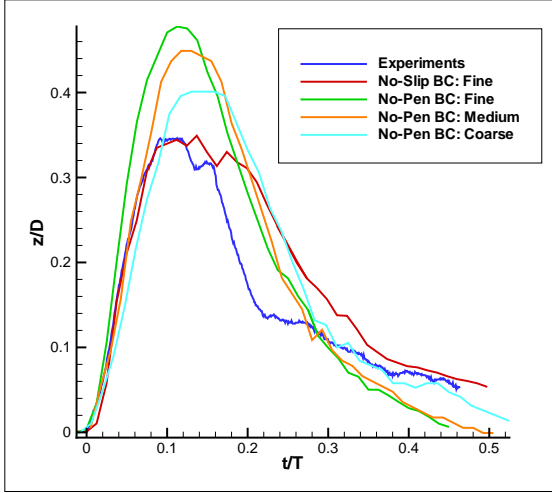


Figure 12: Experimental and simulated wave maker free surface contact lines. Coarse, medium and fine background grids use spacings of $\Delta x = 0.04m, 0.028m$ and $0.02m$ respectively.

breaking events. Note that the two-dimensional nature of this simulation does not allow the breaking wave to fully close and the plunger instead skips off of the free surface.

Because the no-penetration condition has been used, the point of contact between the free surface and the wave maker moves freely in this simulation, raising nearly $0.4m$ at it's peak. Figure 12 shows the experimental measurements of the point of contact on the wave maker surface along with four sets of simulated results. The free surface run-up is denoted (z) and is normalized by the wave maker depth (D). The figure shows that all of the simulations have good general correlation with the experimental measurements. In particular, the no-penetration conditions accurately predict the rates of run-up and run-down compared to the measurements for all resolution levels, but they overshoot the maximum height substantially. In contrast the no-slip condition accurately predicts the run-up and maximum height but the run-down is much too slow, leaving the hull wetted for longer than the experimental result. The no-slip result shown in Figure 12 is for the finest grid only. The coarse and medium grid simulations did not sufficiently resolve the near-wall flow to give accurate results and have not been shown.

The results of Figure 12 and practical limitations on computational resources suggest that a slip model must

be added to the no-penetration condition to model the effects of near-wall viscosity on this wave maker flow and its three-dimensional counterpart. Adding a slip model for the tangential equations of motion using the analytic development given in the previous sections is straightforward and the momentum equation takes on the form

$$\frac{\partial \vec{u}}{\partial t} = (1-\delta') \left(\vec{r} - \frac{1}{\rho} \vec{\nabla} p \right) + \delta' \mathbf{N} \frac{\partial \vec{U}}{\partial t} + \delta' (\mathbf{I} - \mathbf{N}) \vec{K}. \quad (37)$$

\vec{K} is the user defined slip model which can be set as a function of fluid velocity, pressure, density, wall roughness, and any other pertinent parameters. (37) is the most general slip model formulation with proper choice of \vec{K} recovering the no-penetration and no-slip conditions and many possibilities between.

A promising compromise between the no-penetration and no-slip condition is to use a body force correction similar to that developed in Dommermuth et al. (1998) to model the tangential flow. In that formulation, \vec{K} would reintroduce the tangential momentum equations and include a body force term proportional to the slip velocity at the body surface. The force is scaled by a friction coefficient and in Dommermuth et al. (1998) that coefficient is set very high to enforce the no-penetration condition on the hull. However, used in the framework of (37) the coefficient could be tuned based on experimental evidence such as shown in Figure 12. More detailed experimental measurements and numerical studies will allow further investigation of this slip condition, with the goal of a simple and general formulation for free-surface flows.

Another serious modeling concern which must be addressed is the strictly two-dimensional divergence-free air flow. Unlike potential flow simulations, our VOF method models the air flow in addition to the water flow and air trapped inside two-dimensional breaking waves has nowhere to escape. The breaking wave in Figure 11 displays this effect, and does not fully close but skips off of the free surface. In three-dimensions, the majority of air escapes from within a collapsing breaking wave by moving much faster than the bulk flow in the longitudinal direction. This violates the fundamental 2D+t assumptions and models accounting for this effect must be developed to allow 2D+t simulations to be accurately extended to their three-dimensional analogs.

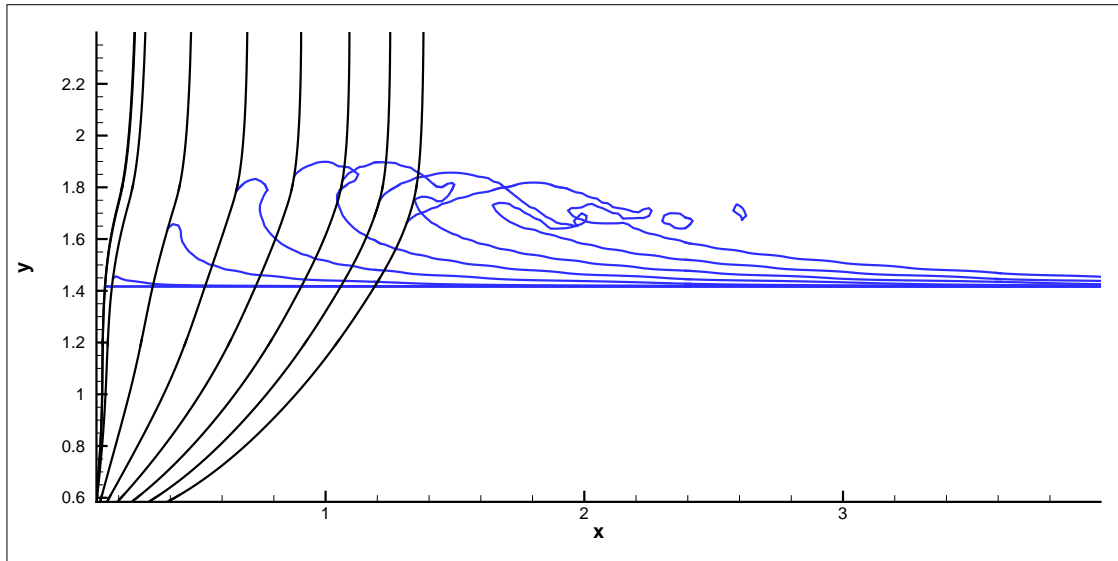


Figure 11: Multiple exposure image of the flexible wave maker and calculated free surface on the finest grid. The black lines are the wave maker surface, and the blue lines are the contours of $f = 0.5$. The no-penetration condition is enforced on the body.

CONCLUSION

This work has shown that Cartesian-grid methods are capable simulating flows with engineering applications without the difficulties associated with fitted-grid methods. New capabilities have been developed to further extend the usefulness and generality of Cartesian grid methods. A free surface advection algorithm which conserves the volume of fluid at the interface even for complex three-dimensional flows has been presented. Exit boundary conditions for the velocity and pressure have been derived which allow for accurate simulation of general external flows. A body boundary formulation based on the analytic alteration of the governing equations has been presented which allows enforcement of general boundary conditions and is easily implemented such that the solver order of accuracy is maintained. A series of simulations of a flexible wave maker were presented which used all of these numerical advancements showed good comparison to experimental results. Specifically, the no-slip and no-penetration boundary conditions each captured elements of the physical system, and a more advanced tangential slip model will help improve the comparison further. An outline of a method to quantify

the bounds on the 2D+t slenderness assumptions were made and the need for a method to allow variation of flow in the air was introduced. With the capabilities presented in this paper highly accurate simulations of general ship flows are achievable.

This work was supported by the Office of Naval Research under grant N00014-01-1-0124 through Dr. L. Patrick Purtell. The computational resources for this work were provided through a Challenge Project grant from the Department of Defense High Performance Computing Modernization Office (Project C1V). The authors wish to thank J. Duncan and M. Shakeri for the use of the experimental wave maker data.

REFERENCES

- Aulisa, E., Manservigi, S., Scardovelli, R., and Zaleski, S., "A geometrical area-preserving Volume-of-Fluid advection method," *J. Comp. Phys.*, 2003, vol. 192, pp. 355–364.
- Dommermuth, D.G., Innis, G., Luth, T., Novikov, E., Schlageter, E., and Talcott, J., "Numerical simulation of bow waves," *22nd Symposium on Naval Hydrodynamics*.

- Dommermuth, D.G., O'Shea, T.T., Wyatt, D.C., Sussman, M., Weymouth, G.D., Yue, D.K., Adams, P., and Hand, R., "The numerical simulation of ship waves using cartesian-grid and volume-of-fluid methods," 26th Symposium on Naval Hydrodynamics.
- Dommermuth, D.G., Sussman, M., Beck, R.F., O'Shea, T.T., Wyatt, D.C., Olson, K., and MacNeice, P., "The numerical simulation of ship waves using cartesian grid methods with adaptive mesh refinement," 25th Symposium on Naval Hydrodynamics.
- Dommermuth, D.G., Sussman, M., O'Shea, T.T., and Wyatt, D.C., "The Numerical Simulation of Ship Waves using Cartesian Grid and Volume-of-Fluid Methods," 26th International Conference Numerical Ship Hydrodynamics.
- Ferziger, J. and Peric, M., Computational Methods for Fluid Dynamics, Springer, 3rd ed., 2002.
- Fontaine, E. and Cointe, R., "A Slender Body Approach to Nonlinear Bow Waves," Phil. Trans. Royal Society of London, 1997, vol. A 355, pp. 565–574.
- Fontaine, E., Faltinsen, O.M., and Coite, R., "New insight into the generation of ship bow waves," J. Fluid Mech., 2000, vol. 421, p. 1538.
- Gilmanov, A. and Sotiropoulos, F., "A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies," J. Comp. Phys., 2005, vol. 207, pp. 457–492.
- Givoli, D., Neta, B., and Patlashenko, I., "Finite element analysis of time-dependent semi-infinite wave-guides with high-order boundary treatment," J. Comput. Phys., 2003, vol. 58, pp. 1955–1983.
- Hendrickson, K. and Yue, D.K.P., "Navier-Stokes simulations of unsteady small-scale breaking waves at a coupled air-water interface," 26th Symposium on Naval Hydrodynamics.
- Higdon, R., "Radiation boundary conditions for dispersive waves," SIAM J. Numerical Analysis, 1994, vol. 31, pp. 24–462.
- Hurt, C.W., "Addition of Wave Transmitting Boundary Conditions to the FLOW-3D Program," 1999, vol. FSI-99-TN49.
- Orlanski, I., "A Simple Boundary Condition for Unbounded Hyperbolic Flows," J. Comp. Phys., 1976, vol. 21, p. 251.
- Peskin, C.S., "The immersed boundary method," Acta Numerica, 2002, pp. 1–39.
- Pilliod, J. and Puckett, E., "Second-order accurate volume-of-fluid algorithms for tracking material interfaces," J. Comput. Phys., 2004, vol. 199, pp. 465–502.
- Puckett, E., Almgren, A., Bell, J., Marcus, D., and Rider, J., "A high-order projection method for tracking fluid interfaces in variable density incompressible flows," J. Comput. Phys., 1997, vol. 130(2), pp. 269–282.
- Rider, W.J. and Kothe, D.B., "Reconstructing Volume Tracking," J. Comp. Phys., 1998, vol. 141, pp. 112–152.
- Scardovelli, R. and Zaleski, S., "Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids," J. Comp. Phys., 2000, vol. 164, pp. 228–237.
- Scardovelli, R. and Zaleski, S., "Interface reconstruction with least-square fit and split EulerianLagrangian advection," Int. J. Numer. Meth. Fluids, 2003, vol. 41, p. 251274.
- Shakeri, M., An experimental 2D+T investigation of breaking bow waves, PhD dissertation, Department of Mechanical Engineering, University of Maryland, 2005.
- Tseng, Y. and Ferziger, J., "A ghost-cell immersed boundary method for flow in complex geometry," J. Comp. Phys., 2003, vol. 192, p. 593623.
- Udaykumar, H.S., Mittal, R., Rampunggoon, P., and Khanna, A., "A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries," J. Comp. Phys., 2001, vol. 174, pp. 345–380.
- Weymouth, G., Hendrickson, K., O'Shea, T., Dommermuth, D.G., Yue, D.K., Adams, P., and Hand, R., "Modeling Breaking Ship Waves for Design and Analysis of Naval Vessels," DOD HPCMP Users Group Conference 2006.
- Ye, T., Mittal, R., Udaykumar, H.S., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," J. Comp. Phys., 1999, vol. 156, pp. 209–240.