24^{th} Symposium on Naval Hydrodynamics
Fukuoka, Japan, 8-13 July 2002

Unstructured Nonlinear Free Surface Simulations for the Fully-Appended DTMB Model 5415 Series Hull Including Rotating Propulsors

Clarence O. E. Burg, Kidambi Sreenivas, Daniel G. Hyams, Brent Mitchell
(Computational Simulation and Design Center, Mississippi State, MS)

ABSTRACT

Nonlinear free surface simulations around realistic geometries, such as the DTMB Model 5415 Series hull, are a necessary step to achieve the goal of simulation of maneuvering surface vessels. As part of this effort, researchers in the Computational Simulation and Design Center at Mississippi State University have developed and parallelized a three-dimensional unstructured code, U^2NCLE, which solves the incompressible Reynolds-averaged Navier-Stokes equations using a node-based flux differencing finite volume method with Roe-averaged variables. The simulations produced by this code realistically capture turbulent flow and vortices arising from bulbous bows and tips of propulsors and rudders and can model the rotation of the propulsors accurately. Because the code uses unstructured grids, the code is applicable to a wide range of geometries, including those with shafts and struts, and can solve the flow through small gaps such as those between the rudder and the boot. Current research efforts include the addition of nonlinear free surface capabilities to this code. A nonlinear free surface is obtained by solving the kinematic free surface boundary condition, imposing the resulting hydrostatic pressure distribution onto the boundary for the Navier-Stokes solver and moving the grid to match the free surface while conforming to any geometry that intersects the free surface. The current free surface implementation is loosely coupled with the Navier-Stokes solver and makes the assumption that the flow is steady-state.

INTRODUCTION

Modeling the free surface generated by surface ships and by submerged vessels near the water surface is important for proper understanding of the flow around these vessels. In particular, the free surface changes the resistance of the vessels through the water by changing the pressure distribution and the wetted surface area. The free surface affects the location and magnitude of the vortices that originate from various locations, including the bow, appendages and propulsors, which can greatly affect the performance of the propulsors. The wave signatures of ships and submarines is also of importance. Surface vessels under investigation by ship designers attempt to mitigate the negative influences of the free surface interaction on cavitation, power requirements and wave signature and attempt to use the free surface to improve the performance in other ways. Because of the novelty of many of these designs, which often are not direct extensions of current designs, extensive sets of experimental data may not yet exist, and designers are more apt to rely on data from numerical simulations. Furthermore, the effects on the ship's performance caused by maneuvering through a change in direction, or through various sea states, especially in regards to sea-keeping and cavitation in littoral water are of importance in the designs and can not be readily tested by experiment.

A long-term goal of the research into numerical simulations is to develop the ability to study the performance of a full ship design, including the interactions of the various appendages, sonar domes, rudders, shafts and propulsors, as the ship maneuvers in response to changes in the settings of the rudders and propulsors. To accomplish this goal, efficient and accurate free surface simulations are needed. Other necessary components to achieve this goal include capabilities to monitor the onset of incipient cavitation, to perform simulation of several vessels in motion relative to one another and to adapt the grid to capture vortices in the water.

Currently, the implicit unstructured code being developed by researchers in the Computational Simulation and Design Center at Mississippi State University solves the three-dimensional, incompressible Navier-Stokes equations, via an edge-based, flux-differencing finite volume method with Roe-averaged variables. Hyams (2000a, 2000b) successfully parallelized the code and has ported the code to a wide variety of high-performance machines. The unstructured grid generator (Marcum, 1998) has the capability of building highly stretched high-aspect ratio grids near viscous surfaces, which include higher order elements such as pyramids and prisms. Other areas of code development include realistic rudder and propulsor-induced motion and full-scale Reynolds number simulation.

At each time step, the nonlinear free surface algorithm solves the kinematic free surface equation

$$\frac{\partial Y}{\partial t}+(u-V_x)\frac{\partial Y}{\partial x}-(v-V_y)+(w-V_z)\frac{\partial Y}{\partial z}=0 \quad (1)$$

where $Y = Y(x,z,t)$ is the free surface defined as a single-valued function over the xz plane, (u,v,w) are the velocity components in the coordinate directions and (V_x, V_y, V_z) are the grid velocities. A flow-through boundary condition for the free surface based on characteristic variable boundary conditions is used, with the pressure on the free surface set to $P = \frac{Y}{Fr^2}$ where P is the pressure and Fr is the Froude number. After several time steps, the grid is moved to match the free surface while conforming to the geometry, using a three dimensional extension of Farhat's torsional spring method (Farhat, 1998). This grid movement algorithm is quite robust, allowing for moderate to extreme distortions as required by the free surface simulations. As the solution converges, the flow along the free surface becomes tangent to the free surface, and the nonlinear free surface solution is obtained.

Much of this work is an extrapolation of the free surface algorithm used within the three-dimensional structured code UNCLE also developed at Mississippi State University. Beddhu (1994)developed the structured free surface solver, using a modified artificial compressibility formulation. This algorithm was applied to steady and unsteady flow around the Wigley hull (Beddhu, 1998a), to the barehull model 5415 series hull (Beddhu, 1998b), to the Series 60 $C_B = 0.6$ ship (Beddhu, 1998c) and to a more detailed stern analysis for the model 5415 series hull (Beddhu, 1999), and these results were presented at the Gothenburg conference (Beddhu, 2000). Initial verification and validation exercises of the unstructured nonlinear free surface algorithm has been presented by Burg (2002), which includes a grid refinement study for flow over a submerged NACA0012 hydrofoil, flow around inviscid and viscous Wigley hulls, and flow around the barehull model 5415 series hull.

In this paper, results for the DTMB Model 5415 Series hull, which has a transom stern, are presented. The numerical results are compared against experimental results which include the hull profile for the barehull 5415 and the stern region topologies for the unpowered and powered fully-appended 5415. Because of the use of unstructured grids, the actual geometry with the shafts, struts, rudder, the gap between the rudder and the boot and the DTMB propellers 4876 and 4877 is used to generate the grid. For the unpowered case, the propulsors are removed, so a steady-state simulation is obtained; for the powered case, the propulsor is rotating providing an unsteady simulation. Even though the nonlinear free surface implementation assumes a steady-state flow, it is assumed that the unsteadiness produced by the propulsor can be adequately modeled under these assumptions.

In the following section, the free surface algorithm is presented, which includes the method for solving the kinematic free surface equation, the imposition of the hydrostatic pressure on the Navier-Stokes equations and the grid movement algorithm. Then, the results for the DTMB Model 5415 series simulations are presented.

NAVIER-STOKES SOLUTION ALGORITHM

Three different sets of equations are solved in the process of simulating the flow around surface vessels and submerged vessels near the water surface. The unsteady incompressible Reynolds-averaged Navier-Stokes equations, which are presented here in Cartesian coordinates and in conservative form, are solved to determine the velocity and pressure within the computational domain. Either the Spalart-Allmaras or the $q-\omega$ turbulence model is used to simulate the turbulent viscosity primarily within the boundary layer, and the kinematic free surface equation is solved to advance the free surface in time.

GOVERNING EQUATIONS

Assuming that gravity acts in the y-direction (i.e., the vertical direction), the incompressible Navier-Stokes equations can be expressed in dimensional form, denoted with superscript *, as

$$\begin{aligned}
&\frac{\partial u^*}{\partial x^*}+\frac{\partial v^*}{\partial y^*}+\frac{\partial w^*}{\partial z^*}=0\\
&\frac{\partial u^*}{\partial t^*}+\frac{\partial}{\partial x^*}\left(u^{*2}+\frac{P^*}{\rho_\infty}\right)+\frac{\partial}{\partial y^*}\left(u^*v^*\right)\\
&\qquad+\frac{\partial}{\partial z^*}\left(u^*w^*\right)=\mu^*\nabla^2u^*\\
&\frac{\partial v^*}{\partial t^*}+\frac{\partial}{\partial x^*}\left(u^*v^*\right)+\frac{\partial}{\partial y^*}\left(v^{*2}+\frac{P^*}{\rho_\infty}+gy^*\right)\\
&\qquad+\frac{\partial}{\partial z^*}\left(v^*w^*\right)=\mu^*\nabla^2v^*\\
&\frac{\partial w^*}{\partial t^*}+\frac{\partial}{\partial x^*}\left(u^*w^*\right)+\frac{\partial}{\partial y^*}\left(v^*w^*\right)\\
&\qquad+\frac{\partial}{\partial z^*}\left(w^{*2}+\frac{P^*}{\rho_\infty}\right)=\mu^*\nabla^2w^* \qquad (2)
\end{aligned}$$

The variables in the preceding equation are normalized with respect to a characteristic length scale (L) and free stream values of velocity (U_∞), density (ρ_∞), and viscosity (μ_∞). Thus, the Reynolds number is defined as $Re = U_\infty L/\nu_\infty$. Pressure is normalized with $P = (P^* + \rho_\infty g y^* - P_\infty)/\rho_\infty U_\infty^2$, where P^* is the local dimensional static pressure. Following (Chorin, 1967), an artificial time derivative term ($\partial\rho_a/\partial t$, where $\rho_a = P/\beta$) has been added to the continuity equation to cast the complete set of governing equations into a time-marching form. The

nondimensionalized equations can be written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} Q d\mathcal{V} + \int_{\partial\Omega} \vec{F}\cdot\hat{\vec{n}} dA = \frac{1}{Re}\int_{\partial\Omega} \vec{G}\cdot\hat{\vec{n}} dA \quad (3)$$

where $\hat{\vec{n}}$ is the outward pointing unit normal to the control volume $\mathcal{V}$. The vector of dependent variables and the components of the inviscid and viscous flux vectors are given as

$$Q = \begin{bmatrix} P \\ u \\ v \\ w \end{bmatrix} \quad (4)$$

$$F\cdot\hat{\vec{n}} = \begin{bmatrix} \beta(\Theta - a_t) \\ u\Theta + \hat{n}_x P \\ v\Theta + \hat{n}_y P \\ w\Theta + \hat{n}_z P \end{bmatrix} \quad (5)$$

$$G\cdot\hat{\vec{n}} = \begin{bmatrix} 0 \\ \hat{n}_x\tau_{xx} + \hat{n}_y\tau_{xy} + \hat{n}_z\tau_{xz} \\ \hat{n}_x\tau_{yx} + \hat{n}_y\tau_{yy} + \hat{n}_z\tau_{yz} \\ \hat{n}_x\tau_{zx} + \hat{n}_y\tau_{zy} + \hat{n}_z\tau_{zz} \end{bmatrix} \quad (6)$$

where β is the artificial compressibility parameter (typically 15 in this work), u, v, and w are the Cartesian velocity components in the x, y, and z directions, and $\hat{n}_x$, $\hat{n}_y$, and $\hat{n}_z$ are the components of the normalized control volume face vector. Θ is the velocity normal to a control volume face:

$$\Theta = \hat{n}_x u + \hat{n}_y v + \hat{n}_z w + a_t \quad (7)$$

where the grid speed $a_t = -(V_x\hat{n}_x + V_y\hat{n}_y + V_z\hat{n}_z)$. Note that $\vec{V}_s = V_x\hat{i} + V_y\hat{j} + V_z\hat{k}$ is the control volume face velocity. The viscous stresses given in Equation 6 are

$$\tau_{ij} = (\mu + \mu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (8)$$

where μ and μ_t are the molecular and eddy viscosities, respectively.

NUMERICAL APPROACH FOR NAVIER-STOKES EQUATIONS

The baseline flow solver is a node-centered, finite volume, implicit scheme applied to general unstructured grids with nonsimplical elements. The flow variables are stored at the vertices and surface integrals are evaluated on the median dual surrounding each of these vertices. The nonoverlapping control volumes formed by the median dual completely cover the domain, and form a mesh that is dual to the elemental grid. Thus, a one-to-one mapping exists between the edges of the original grid and the faces of the control volumes.

The solution algorithm consists of the following basic steps: reconstruction of the solution states at the control volume faces, evaluation of the flux integrals for each control volume, and the evolution of the solution in each control volume in time.

RECONSTRUCTION

A higher order spatial method is constructed by extrapolating the solution at the vertices to the faces of the surrounding control volume. The unweighted least squares method (solved via QR factorization (Anderson, 1994)) is used to compute the gradients at the vertices for the extrapolation. With these gradients known, the variables at the interface are computed with a first order Taylor series as:

$$Q_f = Q_0 + \nabla Q_0 \cdot \vec{r} \quad (9)$$

where $\vec{r}$ is a vector that extends from node 0 to the midpoint of the edge associated with the control volume face in question.

RESIDUAL EVALUATION

The governing equations are discretized using a finite volume technique; thus, the surface integrals in Equation 3 are approximated by a quadrature over the surface of the control volume of interest. So, the numerical discretization of the spatial terms associated with the control volume surrounding vertex 0 results in

$$\frac{\partial q_0}{\partial t} + \Re_0 = 0 \quad (10)$$

where the spatial residual $\Re$ contains all contributions from the discrete approximation to the inviscid and viscous terms ($\Re = \Re_{inv} + \Re_{vis}$). Also, the quantity q is defined as $q = \int_\Omega Q d\mathcal{V}$.

SPATIAL RESIDUAL

The evaluation of the discrete residual is performed separately for the inviscid and viscous terms given in Equation 3. The Roe scheme (Roe, 1981) is used to evaluate the inviscid fluxes at each face of the control volume. The algebraic flux vector is replaced by a numerical flux function, which depends on the reconstructed data on each side of the control volume face:

$$\Phi = \frac{1}{2}\left(F(Q_L) + F(Q_R)\right) - \frac{1}{2}\tilde{A}\left(Q_R - Q_L\right) \quad (11)$$

where $\tilde{A} = \tilde{R}\tilde{\Lambda}\tilde{R}^{-1}$. The matrix R is a matrix constructed from the right eigenvectors of the flux Jacobian, and $\tilde{\Lambda}$ is a diagonal matrix whose entries contain the absolute values of the eigenvalues of the flux Jacobian. The eigensystem used in this work is based on that reported in (Taylor, 1991). Note that $\tilde{A}$ is evaluated with Roe-averaged

variables, which is simply the arithmetic average between left and right solution states in the case of incompressible flows.

For general element grids, it is expedient to use only edge-local information to compute the viscous fluxes. This allows the evaluation of viscous fluxes on each face of the control volume without regard to the varying element types of the mesh. An algorithm in which no element information is used outside of metric computations is termed a "grid transparent" algorithm (Haselbacher, 1999). To this end, the viscous fluxes are evaluated directly at each edge midpoint using separate approximations for the normal and tangential components of the gradient vector to construct the velocity derivatives (Marcum, 1997). Using a directional derivative along the edge to approximate the normal component of the gradient and the average of the nodal gradients to approximate the tangential component of the gradient leads to the following expression (Hyams, 2000b):

$$\nabla Q_{ij} \approx \overline{\nabla Q} + \left[Q_j - Q_i - \overline{\nabla Q} \cdot \vec{\Delta s} \right] \frac{\vec{\Delta s}}{|\vec{\Delta s}|^2} \tag{12}$$

The weighted least squares method is used to evaluate the nodal gradients in the preceding formula.

TEMPORAL RESIDUAL

After the spatial terms have been suitably discretized, the time derivative term appearing in Equation 10 must be approximated. A general difference expression is available for this purpose (Beam, 1978) (Taylor, 1993), and is given as follows:

$$\begin{aligned} \Delta q^n = & \frac{\theta_1 \Delta t}{1+\theta_2} \frac{\partial (\Delta q^n)}{\partial t} \\ & + \frac{\Delta t}{1+\theta_2} \frac{\partial (q^n)}{\partial t} + \frac{\theta_2}{1+\theta_2} \Delta q^{n-1} \end{aligned} \tag{13}$$

where $\Delta q^n = q^{n+1} - q^n$. A first order accurate in time Euler implicit scheme is given by the choices $\theta_1 = 1$, $\theta_2 = 0$. Correspondingly, a second order time accurate Euler implicit scheme is given by $\theta_1 = 1$, $\theta_2 = 1/2$. Since $\theta_1 = 1$ for both time discretizations used in this work, Equation 13 can be further simplified:

$$\Delta q^n = \frac{\Delta t}{1+\theta_2} \frac{\partial}{\partial t} \left(q^{n+1} \right) + \frac{\theta_2}{1+\theta_2} \Delta q^{n-1} \tag{14}$$

Using Equation 10 to replace the time derivative,

$$\frac{\Delta q^n - \frac{\theta_2}{1+\theta_2} \Delta q^{n-1}}{\Delta t} = -\frac{1}{1+\theta_2} \Re^{n+1} \tag{15}$$

By the definition of q, one can write $q = \bar{Q}\mathcal{V}$, where $\bar{Q}$ is the volume averaged solution variable vector $1/\mathcal{V} \int_{\partial\Omega} Q d\mathcal{V}$. Then, the following two identities can be formed:

$$\Delta q^n = \mathcal{V}^{n+1} \Delta \bar{Q}^n + \bar{Q}^n \Delta \mathcal{V}^n \tag{16}$$

$$\Delta q^{n-1} = \mathcal{V}^{n-1} \Delta \bar{Q}^{n-1} + \bar{Q}^n \Delta \mathcal{V}^{n-1} \tag{17}$$

Inserting the above two identities into Equation 15, one arrives at the following expression:

$$\begin{aligned} & \frac{\mathcal{V}^{n+1} \Delta \bar{Q}^n - \frac{\theta_2}{1+\theta_2} \mathcal{V}^{n-1} \Delta \bar{Q}^{n-1}}{\Delta t} \\ & + \bar{Q}^n \left[\frac{\Delta \mathcal{V}^n - \frac{\theta_2}{1+\theta_2} \Delta \mathcal{V}^{n-1}}{\Delta t} \right] \\ & + \frac{1}{1+\theta_2} \Re^{n+1} = 0 \end{aligned} \tag{18}$$

Now, one must consider the Geometric Conservation Law (GCL). This statement relates the rate of change of a physical volume to the motion of the volume faces:

$$\frac{\partial \mathcal{V}}{\partial t} = \int_\Omega \nabla \cdot \vec{V}_s d\mathcal{V} = \int_{\partial\Omega} \vec{V}_s \cdot \hat{\vec{n}} dA \tag{19}$$

According to Thomas and Lombard (1978) and later Janus (1989), the solution of the volume conservation equation must be performed in exactly the same manner as the flow equations to ensure that GCL is satisfied. This procedure ensures that spurious source terms caused by volume changes are eliminated. Using the same time differencing expression (Equation 14) to approximate Equation 19,

$$\frac{\Delta \mathcal{V}^n - \frac{\theta_2}{1+\theta_2} \Delta \mathcal{V}^{n-1}}{\Delta t} = \frac{1}{1+\theta_2} \Re_{\mathrm{GCL}}^{n+1} \tag{20}$$

where $\Re_{\mathrm{GCL}}^{n+1} = \sum_{i \in \mathcal{N}(0)} \vec{V}_{s,0i}^{n+1} \cdot \vec{n}_{0i}^{n+1}$, and $\mathcal{N}(0)$ is a list of nodes surrounding node 0. Note that the left hand side of the preceding equation is exactly the bracketed term in Equation 18. Replacing the bracketed term and rearranging slightly gives the final form of the discretization of the time derivative:

$$\begin{aligned} & \frac{(1+\theta_2)\mathcal{V}^{n+1} \Delta \bar{Q}^n - \theta_2 \mathcal{V}^{n-1} \Delta \bar{Q}^{n-1}}{\Delta t} + \\ & \bar{Q}^n \Re_{\mathrm{GCL}}^{n+1} + \Re^{n+1} = 0 \end{aligned} \tag{21}$$

For incompressible flows, a divergence-free velocity field at the end of each Newton iteration is desirable. To this end, the contribution of the time derivative and GCL terms to the residual are removed for the continuity equation.

TIME EVOLUTION

A Newton iterative time evolution scheme is applied, which requires the solution of a sparse linear system at each nonlinear subiteration:

$$-\Im_0^{n+1,m} = \left. \frac{\partial \Im_0}{\partial \bar{Q}} \right|^{n+1,m} \Delta \bar{Q}^{n+1,m} \tag{22}$$

where

$$\begin{aligned} & \Im_0^{n+1} \left(\bar{Q}^{n+1} \right) \\ & = \frac{(1+\theta_2)\mathcal{V}_0^{n+1} \Delta \bar{Q}_0^n - \theta_2 \mathcal{V}_0^{n-1} \Delta \bar{Q}_0^{n-1}}{\Delta t} \\ & + \bar{Q}^n \Re_{0,\mathrm{GCL}}^{n+1} + \Re_0^{n+1} \end{aligned} \tag{23}$$

where $\Delta\bar{Q}^{n+1,m} = \bar{Q}^{n+1,m+1} - \bar{Q}^{n+1,m}$. Now, expanding the terms and performing the required differentiations of $\Im$ results in the following expression for Newton's method:

$$-\left[\frac{(1+\theta_2)\mathcal{V}_0^{n+1}(\bar{Q}_0^{n+1,m}-\bar{Q}_0^n)-\theta_2\mathcal{V}_0^{n-1}\Delta\bar{Q}_0^{n-1}}{\Delta t}\right.$$

$$\left.+\bar{Q}_0^n\Re_{0,\mathrm{GCL}}^{n+1}+\sum_{i\in\mathcal{N}(0)}\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}\right]=$$

$$\left[\frac{(1+\theta_2)\mathcal{V}_0^{n+1}I}{\Delta t}+\right.$$

$$\left.\sum_{i\in\mathcal{N}(0)}\frac{\partial\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}}{\partial\bar{Q}_0}\right]\Delta\bar{Q}_0^{n+1,m}+$$

$$\sum_{i\in\mathcal{N}(0)}\left[\frac{\partial\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}}{\partial\bar{Q}_i}\Delta\bar{Q}_i^{n+1,m}\right] \qquad (24)$$

where $\Delta\bar{Q}_0^{n-1} = \bar{Q}_0^n - \bar{Q}_0^{n-1}$. For notational convenience, both the inviscid and viscous terms are collapsed into a single flux function H. Note that the iteration can be started by using an initial guess of $\bar{Q}^{n+1,0} = \bar{Q}^n$. Also, performing only one iteration of Newton's method per time step (with 1st order time discretization and no GCL terms) is equivalent to a time linearization of the spatial terms only. However, writing the method in this framework is more general than a straightforward time linearization of the nonlinear terms.

To solve the resulting linear system, a bidirectional Gauss-Seidel solution algorithm is adopted. Splitting the matrix into diagonal, upper triangular, and lower triangular parts as $[\mathcal{A}] = [\mathcal{L}]+[\mathcal{D}]+[\mathcal{U}]$, the Gauss-Seidel sweeps may be written as the following two-step process per subiteration (k is the linear subiterative index):

$$[\mathcal{L}+\mathcal{D}]\,\Delta\bar{Q}^{k+\frac{1}{2}}+[\mathcal{U}]\,\Delta\bar{Q}^{k}=\Re^{n+1,m} \qquad (25)$$

$$[\mathcal{D}+\mathcal{U}]\,\Delta\bar{Q}^{k+1}+[\mathcal{L}]\,\Delta\bar{Q}^{k+\frac{1}{2}}=\Re^{n+1,m} \qquad (26)$$

where it is understood that $\Delta\bar{Q}$ is evaluated at the $n+1, m+1$ time level and Newton subiteration. The diagonal, lower, and upper operators are defined as

$$\mathcal{D}=\left[\frac{(1+\theta_2)\mathcal{V}_0 I}{\Delta t}+\sum_{i\in\mathcal{N}(0)}\frac{\partial\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}}{\partial\bar{Q}_0}\right](\cdot) \qquad (27)$$

$$\mathcal{U}=\sum_{i\in\mathcal{N}_U(0)}\frac{\partial\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}}{\partial\bar{Q}_i}(\cdot) \qquad (28)$$

$$\mathcal{L}=\sum_{i\in\mathcal{N}_L(0)}\frac{\partial\vec{H}_{0i}^{n+1,m}\cdot\vec{n}_{0i}^{n+1}}{\partial\bar{Q}_i}(\cdot) \qquad (29)$$

The initial guess to begin the iteration is $\Delta Q^{n+1,0} = 0$. In the above formulas, $\vec{H}_{0i}$ represents the combined inviscid and viscous flux vector from node 0 to node i. $\mathcal{N}(0)$ represents the set of neighbors for node 0, $\mathcal{N}_L(0)$ is the list of neighbors such that the node label $\ell(i) < \ell(0)$, and $\mathcal{N}_U(0)$ is the list of neighbors such that the node label $\ell(i) > \ell(0)$

BOUNDARY CONDITIONS

Viscous conditions are enforced by modifying the linear system such that no change is allowed in the velocity, and the pressure is driven according to the imbalance in the continuity equation in the boundary control volume (Anderson, 1994). Farfield conditions are handled via a characteristic variable reconstruction; all boundary conditions are handled in an implicit fashion. The free surface boundary conditions will be presented below - in essence, at steady-state, the velocity is tangent to the free surface (i.e., $\vec{u}\cdot\hat{n}_{fs} = 0$) and the pressure is $P = \frac{Y}{Fr^2}$, where Y is the elevation of the free surface above the undisturbed waterline and the Froude number is $Fr = \frac{U_\infty}{\sqrt{gL}}$.

A symmetry plane boundary condition imposes $\nabla\psi\cdot\vec{b} = 0$ for any arbitrary variable ψ. In addition, no flow is allowed through a symmetry surface; so, like a solid wall condition, $\theta = 0$. To simulate symmetry conditions, a layer of phantom entities that is a mirror image of the entities inside and connected to the symmetry plane is created; one element layer is mirrored. Thus, control volumes on the symmetry plane are closed and behave just as interior control volumes. Phantom nodes created by the mirroring process are updated in the same hierarchy as described for phantom nodes generated by the parallelization. Note that care must be taken to copy scalars, mirror vectors, and mirror tensors appropriately to the symmetry phantom nodes such that no fluxes are allowed through the plane.

TURBULENCE MODEL

Both the one-equation turbulence model of Spalart and Allmaras (1992)(Sheng, 1999) and the $q-\omega$ turbulence model (Coakley, 1985) are available within the solver. Constants are taken from the version II $q-\omega$ model given in (Coakley, 1985). The interested reader is referred to these references.

The diffusive terms in both turbulence models are discretized in the same manner as the viscous terms for the mean flow, and the convective terms are computed via pure upwinding. Appropriate consideration is given to maintain positive operators in the formation of the Jacobian matrix for the implicit solution of the transport equation(s). The respective turbulence models are incorporated with the mean flow solution in a "loosely-coupled" procedure; that is, the core governing equations are solved

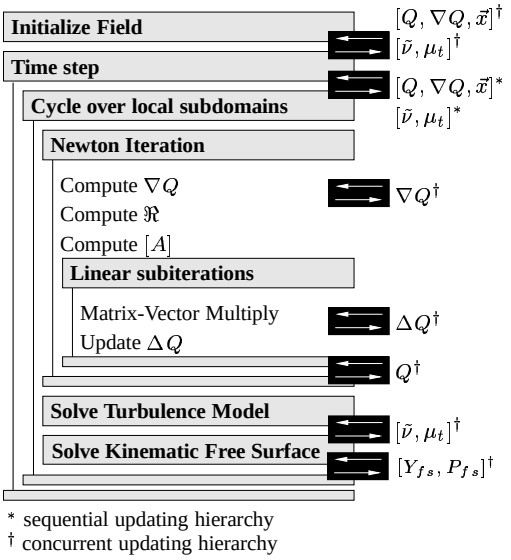

Figure 1: Iteration hierarchy used for the parallel unstructured solution algorithm

first, then the turbulence model is solved independently. This procedure allows for easy interchange of the turbulence models.

PARALLELIZATION

For quick turnaround time in a design environment, it is essential to parallelize the flow solution algorithm. The present parallel unstructured viscous flow solver is based on a coarse-grained domain decomposition for concurrent solution within subdomains assigned to multiple processors. The solution algorithm employs iterative solution of the implicit approximation, with the concurrent iteration hierarchy as shown in Figure 1. Also, domain decomposition takes place with each node in the domain uniquely mapped to a given task. The code employs MPI message passing for interprocessor communication.

In general, the parallelization of an existing validated flow solver should satisfy several constraints. First and most important, the accuracy of the overall numerical scheme must not be compromised; i.e., the solution computed in parallel must have a one-to-one correspondence with the solution computed in serial mode. For the current numerical algorithm, this ability has been shown in (Hyams, 2000). Also, the code must be efficient in its use of computational resources. This characteristic is measured in terms of memory usage and scalability, as well as the fact that the parallel code should degenerate to the serial version if only one processor is available. Finally, the consequences of the inevitable domain decomposition should not seriously compromise the convergence rate of the iterative algorithm.

NONLINEAR FREE SURFACE ALGORITHM

A nonlinear free surface is obtained for a steady-state simulation by solving the kinematic free surface boundary condition at each time level and imposing the hydrostatic pressure distribution based on the new free surface elevation onto the free surface boundary within the mean flow. After several time steps, typically on the order of 200, the grid is moved to match the free surface elevations while conforming to the surfaces that intersect the free surface, such as the hull of a ship or the sail of a submarine; and as the loosely coupled interaction between the free surface, the Navier-Stokes equations and the grid movement algorithm converges, the solution approaches the nonlinear free surface solution.

GOVERNING EQUATIONS

In deriving the kinematic free surface boundary condition, the free surface $\eta(x, y, z, t) = 0$ is considered as a material surface (i.e., no flow travels through the surface), so

$$\frac{D\eta}{Dt} = 0 \tag{30}$$

or

$$\frac{\partial \eta}{\partial t} + (u - V_x)\frac{\partial \eta}{\partial x} + (v - V_y)\frac{\partial \eta}{\partial y} + (w - V_z)\frac{\partial \eta}{\partial z} = 0 \tag{31}$$

By making the assumption that the free surface η can be expressed as $\eta(x, y, z, t) = y - Y(x, z, t) = 0$, this equation reduces to

$$\frac{\partial Y}{\partial t} + (u - V_x)\frac{\partial Y}{\partial x} - (v - V_y) + (w - V_z)\frac{\partial Y}{\partial z} = 0 \tag{32}$$

At steady-state, the kinematic boundary condition becomes

$$\begin{aligned} (\vec{u} + a_t) \cdot \left(\frac{\partial Y}{\partial x}, -1, \frac{\partial Y}{\partial z} \right) &= 0 \\ (\vec{u} + a_t) \cdot \alpha \hat{n}_{fs} &= 0 \end{aligned} \tag{33}$$

where $\alpha = \sqrt{1 + \left(\frac{\partial Y}{\partial x}\right)^2 + \left(\frac{\partial Y}{\partial z}\right)^2}$, so the flow is tangent to the free surface at steady-state.

The pressure in the numerical simulation at the free surface is set based on the free surface elevation. For the original, dimensional equations, the pressure at the free surface is the atmospheric pressure P_∞. Using the nondimensionalization, $P = \frac{P^* + \rho_\infty g y^* - P_\infty}{\rho_\infty U_\infty^2}$ with $P^* = P_\infty$ and $\frac{y^*}{L} = Y$, the pressure is clearly set to $P = \frac{Y}{Fr^2}$, where the Froude number is $\frac{U_\infty}{\sqrt{gL}}$.

NUMERICAL APPROACH FOR FREE SURFACE

The kinematic equation for the linear free surface is solved via a Galerkin approach. In this approach, an algebraic equation is obtained for each node j by multiplying the

governing equation by a weight function $\varphi_j(x,z)$, approximating the terms in the governing equation by using linear and bilinear interpolating functions $\phi_i(x,z)$ and integrating over the computational domain, or

$$\int_\Omega \varphi_j(x,z) \left(\frac{\partial \tilde{Y}}{\partial t} + \left(\tilde{u} - \tilde{V}_x\right) \frac{\partial \tilde{Y}}{\partial x} - \left(\tilde{v} - \tilde{V}_y\right) + \left(\tilde{w} - \tilde{V}_z\right) \frac{\partial \tilde{Y}}{\partial z} \right) = 0 \quad (34)$$

where the interpolated free surface elevation is $\tilde{Y}(x,z) = \sum_i Y_i \phi_i(x,z)$ and the velocities are of the form $\tilde{u}(x,z) = \sum_i u_i \phi_i(x,z)$.

For the triangular elements, the resulting equations are integrated either exactly or using 7 point Gauss quadrature, and for the quadrilateral elements, the integrals are solved either using 3 or 6 point Gauss quadrature in both directions or by decomposing the quadrilaterals into 4 overlapping triangles and evaluating them as triangles. The examples given herein use 7 point Gauss quadrature for the triangular elements and 6 point Gauss quadrature for the quadrilateral elements. A first-order backward time discretization of Equation 34 is used for the temporal derivative, where $\frac{\partial \tilde{Y}}{\partial t} = \frac{\tilde{Y}^{n+1} - \tilde{Y}^n}{\Delta t}$ and the spatial terms evaluated at time level $n+1$. This discretization results in a linear algebraic system of the form $\Re_{fs}(Y^{n+1}, Y^n) = 0$. Since these equations are linear in the unknown variable Y_i^{n+1} at each time level (i.e., the velocities are frozen), the Jacobian matrix is calculated only once per time level. A Gauss-Seidel iterative solver similar to the one used to solve the discretized Navier-Stokes equations is used to identify the solution of the kinematic free surface equation by solving

$$\left[\frac{\partial \Re_{fs}}{\partial Y^{n+1}}\right] \Delta y^{n+1,m} = -\Re_{fs}(Y^{n+1,m}, Y^n) \quad (35)$$

where $Y^{n+1,m+1} = Y^{n+1,m} + \Delta y^{n+1,m}$ and $Y^{n+1} = Y^{n+1,m+1}$ when $\|\Delta y^{n+1,m}\| <$ tolerance.

Within the viscous boundary layer, special care is needed in solving the kinematic free surface equation. On viscous surfaces, the flow velocity plus the grid velocity (i.e., $\vec{u} + a_t$) is set to zero, which prevents the free surface from moving at the viscous surface. Physically, however, surface tension effects force the free surface to rise and fall along the viscous surfaces. Since the solver does not simulate surface tension, another method must be employed to move the free surface near the viscous surfaces. Following the method used in the structured solver, the free surface at a certain distance from the viscous surface is extended at a constant height to the viscous surface. This distance is typically on the order of 2×10^{-4} times the characteristic length L, whereas the point spacing off of the viscous surface is on the order 10^{-6} for model scale simulations and 10^{-8} for full scale simulations.

FREE SURFACE BOUNDARY CONDITION

At the free surface boundary, the pressure is set to the hydrostatic pressure of $P = \frac{Y}{Fr^2}$. Determining the velocity to impose on the boundary is more difficult. If the velocity were forced to be tangent to the free surface (i.e., $(\vec{u} + a_t) \cdot \hat{n}_{fs} = 0$), then the free surface equation would reduce to $\frac{\partial \tilde{Y}}{\partial t} = 0$, and the free surface could not evolve. Thus, this tangency condition is relaxed to allow the free surface to evolve.

This boundary condition is implemented using a characteristic variable boundary condition, similar to the derivation used for the farfield boundary condition. For hyperbolic systems, flow information travels along characteristics; and for three-dimensional incompressible flow, three characteristics originate upstream, and one originates downstream. If the flow is traveling out of the domain (i.e, $(\vec{u} + a_t) \cdot \hat{n} > 0$), then only one characteristic needs to be specified from the outside, which is derived from the hydrostatic pressure. If the flow is traveling into the domain, three characteristics need to be specified from the outside. The characteristic variables are derived from considering the inviscid fluxes across the the boundary face, via

$$\frac{\partial Q}{\partial t} + \nabla F \cdot \hat{\vec{n}} = 0 \quad (36)$$

where $\hat{\vec{n}}$ is the normal to the grid for each boundary face. The inviscid flux term can be rewritten as

$$\nabla F = \frac{\partial F}{\partial Q} \nabla Q = R \Lambda R^{-1} \nabla Q \quad (37)$$

where Λ is a diagonal matrix with the eigenvalues Θ, Θ, $\Theta + c$ and $\Theta - c$, where $\Theta = u\hat{n}_x + v\hat{n}_y + w\hat{n}_z + a_t$ and $c = \sqrt{\left(\Theta - \frac{a_t}{2}\right)^2 + \beta}$. Premultiplying by R^{-1},

$$R^{-1} \frac{\partial Q}{\partial t} + \Lambda R^{-1} \nabla Q \cdot \hat{\vec{n}} = 0 \quad (38)$$

Freezing the matrix $R_o^{-1} = R^{-1}(Q^{in})$,

$$\frac{\partial R_o^{-1} Q}{\partial t} + \Lambda \nabla \left(R_o^{-1} Q\right) \cdot \hat{\vec{n}} = 0 \quad (39)$$

Using the definition $W(Q) = R_o^{-1} Q$, Equation 36 decouples into four equations for the four characteristic variables W as

$$\frac{\partial W}{\partial t} + \Lambda \nabla W \cdot \hat{\vec{n}} = 0 \quad (40)$$

Three of the characteristics are obtained from upstream values, and one of the characteristics is obtained from downstream values. In the case of a boundary face, there are a set of characteristics associated with the inside flow

$W^{in}(Q^{in})$ and a set of characteristics associated with the outside flow $W^{out}(Q^{out})$. The flow variables on the boundary face are determined from the appropriate characteristic variables via $Q = R_o W$. For flow traveling out of the domain, the characteristic variables are chosen to be

$$W = \begin{bmatrix} w_1^{in} \\ w_2^{in} \\ w_3^{in} \\ w_4^{out} \end{bmatrix} \tag{41}$$

because the fourth eigenvalue $\Theta - c$ is negative, while the other eigenvalues are positive. For flow traveling into the domain, only the third eigenvalue is positive, so the characteristic variables are chosen as

$$W = \begin{bmatrix} w_1^{out} \\ w_2^{out} \\ w_3^{in} \\ w_4^{out} \end{bmatrix} \tag{42}$$

For farfield boundary conditions, the characteristic variables associated with the outside flow are taken as the freestream variables. For free surface boundary conditions, the characteristic variables for the outside flow are determined from the hydrostatic pressure and the only available velocity information, which is the velocity on the inside. However, this velocity is modified by removing a portion of the velocity component that is not tangent to the grid, via

$$\vec{u}^{out} = (\vec{u} + a_t) - \gamma \left((\vec{u} + a_t) \cdot \hat{n} \right) (\vec{u} + a_t) \tag{43}$$

where

$$\gamma = \begin{cases} -(\vec{u} + a_t) \cdot \hat{n} & \text{if} \quad -1 \leq (\vec{u} + a_t) \cdot \hat{n} < 0 \\ 1 & \text{if} \quad (\vec{u} + a_t) \cdot \hat{n} < -1 \end{cases} \tag{44}$$

When γ is 1, the velocity imposed from the outside is tangent to the grid (i.e., $\vec{u^{out}} \cdot \hat{n} = 0$). When the velocity was not modified or constrained in some fashion, the free surface algorithm became unstable for flow into the domain due to the inconsistency of using downstream information. This modification provides enough control over the velocity to maintain stability of the algorithm, and if the grid is allowed to move to match the free surface, then this modification ensures that the flow will be tangent to the free surface at convergence.

GRID MOVEMENT ALGORITHM

After several time steps, the grid is moved to match the free surface while conforming to any solid surfaces intersecting the free surface, with displacements on the surface being propagated into the volume grid. Several methods are available for this grid movement, including the use of a Laplacian solver to propagate the surface perturbations into the volume grid, the use of a linear spring analogy where each edge is a spring whose stiffness is determined by the length of the edge, a torsional/linear spring analogy where the angle between the edges affects the stiffness of the springs in the mesh, or solving the linear elasticity equations to propagate the perturbations within the grid. The spring analogy is computationally efficient but is not robust. Both the spring analogy and the Laplacian solver generate elements with negative volumes for moderate amounts of movement, on the order of the size of 3 to 4 elements. Solving the linear elasticity equations to move the grid is robust, but the computational cost associated with this method is the primary drawback.

Thus, following the work of (Farhat, 1998) and extending it to three dimensional grids, the torsional/linear spring analogy has been developed and applied to the problem of moving the grid to match the linear free surface elevations. For tetrahedral grids, this method is quite robust, allowing severe distortions on the surface while maintaining positive volume elements. In practice, this algorithm allows up to approximately 80% compression of elements. Within the boundary layer, (i.e., near the viscous surfaces), the grid moves the same as the points on the boundary. This grid movement method is computationally costly but provides excellent robustness for the free surface solver.

The linear spring method for moving an unstructured grid is presented in (Batina, 1989) where each edge in the grid is replaced by a linear spring whose stiffness is inversely proportional to the length of the edge. Thus, for the edge connecting nodes i and j, the stiffness k_{ij} of the spring is

$$k_{ij} = \frac{1}{((x_i - x_j)^2 + (y_i - y_j)^2)^{p/2}} = \frac{1}{l_{ij}^p} \tag{45}$$

where the coordinates of nodes i and j are (x_i, y_i) and (x_j, y_j) and p is a predetermined coefficient, usually 1 or 2, and $l_{ij} = \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)}$. Given a set of nodal displacements or forces acting on the boundary of the computational grid, the following equations for the interior displacements are solved iteratively until all the forces are in equilibrium

$$\begin{aligned} \Delta x_j^{n+1} &= \frac{\sum_i k_{ij} \Delta x_i^n}{\sum_i k_{ij}} \\ \Delta y_j^{n+1} &= \frac{\sum_i k_{ij} \Delta y_i^n}{\sum_i k_{ij}} \end{aligned} \tag{46}$$

where i is summed over all edges connected to node j. This method is easily extended to three dimensions and is computationally efficient requiring only a few Jacobi iterations to achieve an acceptable level of accuracy. Several researchers have used this method to move nodes in unstructured grids for simulating flows around objects in relative motion (Singh, 1995) and for design optimization (Anderson, 1997).

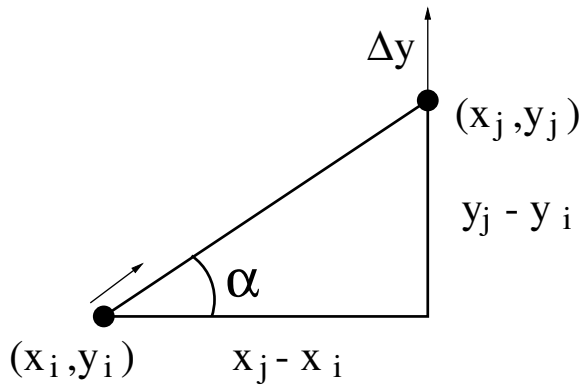

Figure 2: A Displacement in y-direction results in displacements in x and y directions

Technically speaking, these equations do not simulate the behavior of a network of springs because there is no interaction between the x and y coordinates. A displacement in one coordinate will not influence the location in the other coordinate, as would be the case for a network of springs. An example of this coupled interplay is shown in Figure 2, where a displacement in the y-direction is applied to node j which results in a displacement in both directions for node i. To simulate the behavior of a network of springs, the following set of equations for each spring must be summed over all springs:

$$\begin{aligned} &\text{forces in the x-direction :} \\ &\quad k_{ij}\left[(\Delta x_i - \Delta x_j)\cos^2\alpha \right. \\ &\quad \left. + (\Delta y_i - \Delta y_j)\cos\alpha\sin\alpha\right] \\ &\text{forces in the y-direction :} \\ &\quad k_{ij}\left[(\Delta x_i - \Delta x_j)\cos\alpha\sin\alpha \right. \\ &\quad \left. + (\Delta y_i - \Delta y_j)\sin^2\alpha\right] \end{aligned} \tag{47}$$

For either representation, however, this method often fails for complicated geometries and for large changes in the boundary, because negative areas are generated when a node crosses over an edge in the grid. The creation of negative areas is illustrated in Figure 3, where node 1 is pushed downwards and node 4 is forced to cross the edge between nodes 2 and 3. The reason for this failure is that the stiffness in the linear spring method prevents two nodes from colliding but does not prevent a node from crossing over an edge. As two nodes get closer together, the stiffness increases without bound preventing the collision; but there is no mechanism to prevent a node from crossing an edge, because these crossovers can occur without the stiffness increasing without bound.

To provide a more robust movement algorithm for unstructured grid, Farhat (1998) developed an algorithm to prevent a node from crossing an edge by using torsional springs around each node, as shown in Figure 4.

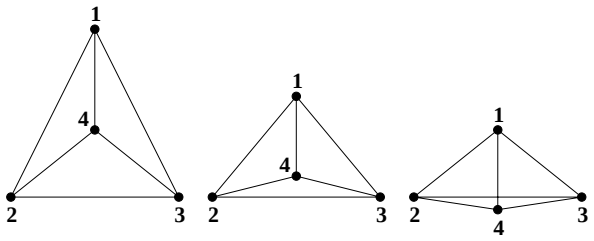

Figure 3: Negative Areas Produced by Linear Spring Method

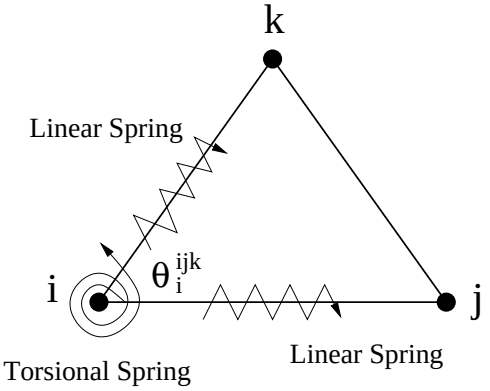

Figure 4: Placement of Linear and Torsion Springs

The stiffness of the torsion spring C_{ijk} is inversely proportional to the sine of the angle, so that the stiffness grows without bound as the angle decreases towards zero or increases towards 180^o, or

$$C_{ijk} = \frac{1}{\sin^2\theta_i^{ijk}} \tag{48}$$

where θ_i^{ijk} is the angle centered at node i formed from the edges ij and ik. Thus, as a node moves towards an edge, the angle goes towards zero, and the stiffness of the torsion spring grows. The sine of the angle is squared to prevent a negative stiffness.

In (Farhat, 1998), the equations for the torsional spring methods are derived for two-dimensional grids, and an iterative solution method is presented. In the present work, these methods have been extended into three dimensions, which provides an adequate level of robustness for the grid movement algorithm. The derivations of the three-dimensional torsional spring equations are beyond the scope of this paper and will be presented elsewhere.

EXAMPLES

The two examples given in this paper deal with the fully-appended DTMB Model 5415 series hull, one without propulsors and the other with propulsors rotating. Experimental data for comparison was provided by Ratcliffe

at the Naval Surface Warfare Center Carderock Division (NSWCCD) (Ratcliffe, 1990, 2000), which consisted primarily of the profile of the free surface intersection with the hull and free surface elevations in the stern of the hull. The Model 5415 series hull has a transom stern, which produces a sizeable "rooster tail" in the water surface behind the stern. The parameters for the various runs are given in Table 1.

Parameter	Value
Speed U_∞ (m/sec)	2.06
Ship Length L (m)	5.72
Propulsor RPM	436
Advance Ratio (Rev. per L)	20.177
Froude Number (Fr)	0.28
Reynolds Number (Re)	11.655 Million
Omega	126.59 rad/second
Δt for 1.5 degrees	2.068089×10^{-4}

Table 1. Flow parameters for Model 5415 Series simulations.

The grids used within the following simulations were generated by AFLR3 (Marcum, 1998) and consist of prisms in the boundary layer generated from the triangles on the viscous surfaces, tetrahedra and pyramids in the transition region at the edge of the boundary layer grid and tetrahedra outside of the boundary layer. On the free surface grid, quadrilaterals exist within the boundary layer region and triangles exist outside of the boundary layer. The use of tightly packed prisms within the boundary layer allows for good resolution of the boundary layer effects and typically generates y^+ values on the order of 1.0.

Information about the size of the grids used in these simulations is provided in Table 2. For the powered and unpowered simulations, the same grid is used except for the regions around the two propulsors. For the unpowered grid, the propulsors were replaced by hubs and were not rotated. For these simulations, the grid included both sides of the hulls, so that maneuvers using these grids could be studied in the future. The hull, struts, rudders and propulsors were symmetric about the centerline, but because of the use of unstructured grids, the grid was not symmetric about the centerline. These simulations were run locally on a LINUX-based cluster running MPI-Pro.

	Unpowered	Powered
Nodes	4251444	5437074
Tetrahedra	8195672	9912263
Pyramids	23592	35101
Prisms	5446868	7160111
Processors	32	42

Table 2: Information about Grids.

In each case, the simulation was run with a constant CFL until the gross features in the flow field developed, after which the simulation was run at a constant time step, so that a time accurate simulation could be achieved for the powered case. To ensure a valid comparison, both simulations were run at a constant time step. For the powered case, the images and plots use instantaneous data rather than time-averaged data.

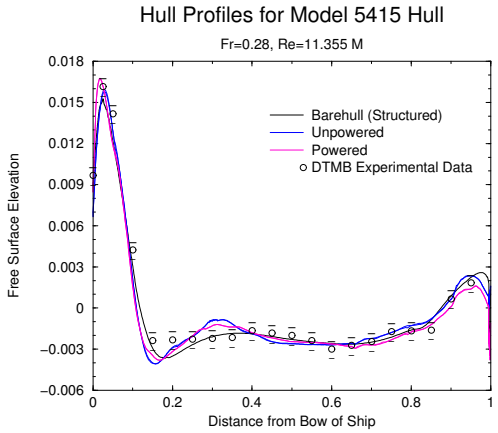

Figure 5: Free Surface / Hull Intersection

UNPOWERED FULLY-APPENDED 5415

The unpowered fully-appended 5415 simulation used the same grid as the powered fully-appended 5415 simulation with the exception that the grid around the propulsor/hub was changed and did not rotate. Hence, the free surface grid was identical and the forces acting on the free surface near the bow should be almost identical between the simulations. The profile of the intersection between the free surface and hull is given in Figure 5. This profile compares the results for the powered and unpowered simulations with the experimental results from the Naval Surface Warfare Center Carderock Division runs (Ratcliffe, 1990, 2000) and with the computational results of the structured flow solver UNCLE (Beddhu, 1999, 2000). The profile for the powered and unpowered cases agree with each other nicely until approximately $y/L = 0.6$ at which point the unsteady influence of the propulsors are felt. The bow wave in the unstructured grids are higher than for the structured grid and seem to agree more with the bow wave elevation of the experiment. The drop behind the bow wave is more severe for the unstructured simulations - this difference may be caused by a lack of refinement in this region.

Figure 6 shows the bow wave for the unpowered simulation head on, as well as the bulbous bow underneath the free surface. In Figure 7, the surface grid near the bow is shown. This region undergoes the largest displacement due to the changes in the free surface, and yet the grid movement algorithm is able to produce nice quality in the

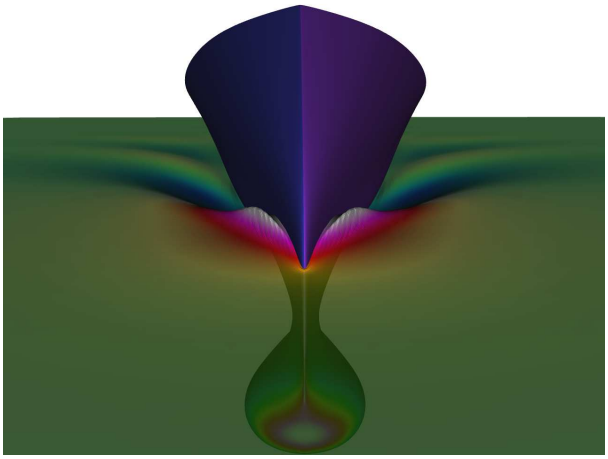

Figure 6: Bow Wave for Model 5415

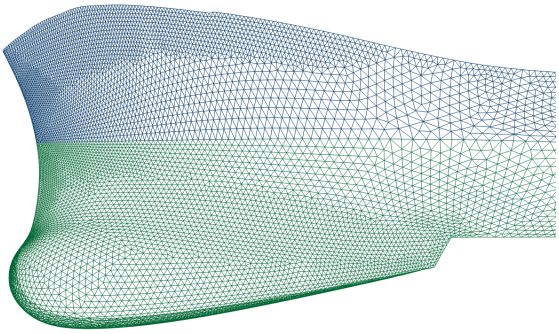

Figure 7: Grid Moved to Match Free Surface Elevation near Bow

distorted grid.

The hullform was symmetric about the centerline, but due to the use of unstructured grids, neither the volume nor the surface grids were symmetric. This asymmetry in the grid produces an asymmetry in the discretization error within the code, but the free surface, even in the stern region, regains symmetry as the solution approaches steady-state. In Figures 8 and 9, contour plots of the free surface elevations around the unpowered 5415 are presented showing the Kelvin wake pattern originating from the bow and the "rooster tail" in the stern region. For the unpowered case, the solution is nearly symmetric. Due to the coarseness of the grid away from the body, the free surface dissipates approximately one body length behind the hull. By extending the resolution on the free surface, the free surface waves could be carried further. The free

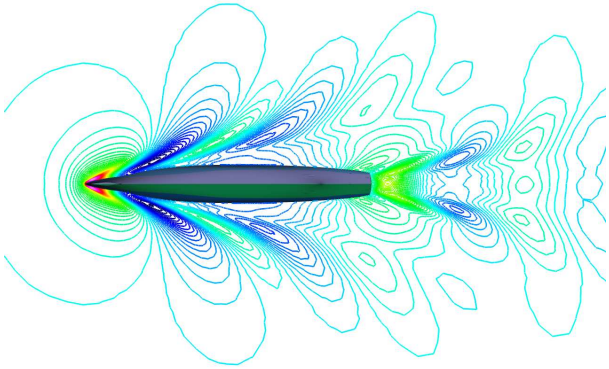

Figure 8: Contours of Free Surface for Unpowered 5415

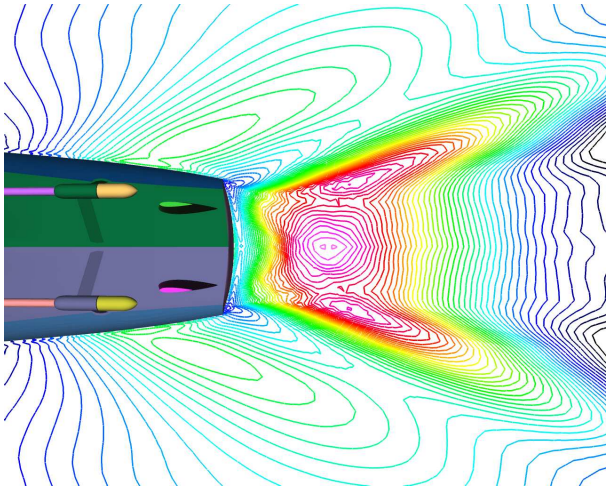

Figure 9: Contours of Free Surface for Stern Region of Unpowered 5415

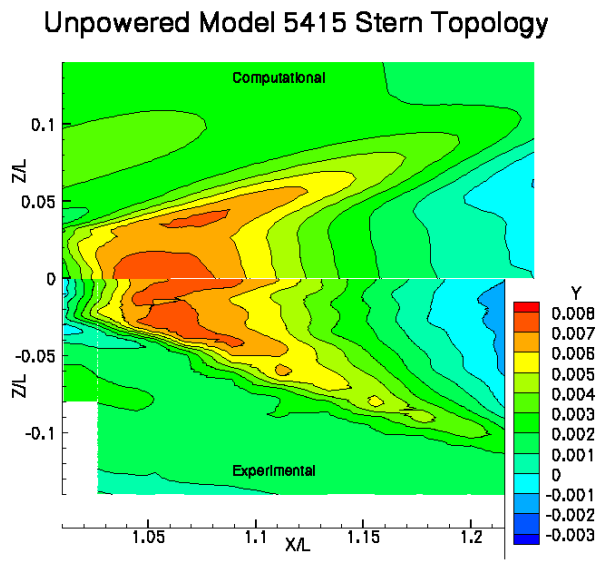

Figure 10: Comparison with Experimental Data for Unpowered 5415

surface in the stern region was also compared with the experimental data from Ratcliffe at NSWCCD and is shown in 10. The agreement for the unpowered case is remarkable, with similar features and elevations. There is a noticeable shift forward for the computational results.

Finally, the free surface-stern intersection for the powered and unpowered simulations are given in Figure 11, as well as the profile for the structured barehull simulation. The structured barehull simulation was run using a symmetry plane, so its profile was mirrored about $z = 0$. The unpowered simulation is almost symmetric about the stern, whereas the unsteadiness in the powered simulation has created some asymmetry (this asymmetry is probably a result of not being as highly converged as the unpowered case). For the powered simulation, this stern profile is an instantaneous cut, and is not as converged as the unpowered simulation. The profiles for the simulations differ greatly. The reason for this difference lies primarily with the difference in the flow field behind the rudders. For the barehull, the struts, shafts and rudder have been removed, so the flow has not been slowed nor straightened, thus the flow passing the stern is faster for the barehull case than for the unpowered case. For the unpowered case, the flow has been slowed by these appendages and straightened by the rudder, so that the flow is slower, and the free surface is not sucked down. And for the powered case, the propulsors have accelerated the flow past the rudders, which sucks the free surface down. In Figure 12, the vertical velocity on the free surface and for a cutting plane just aft of the stern are presented from a view beneath the hull, showing that the velocity is relatively uniform between the rudders and the vertical velocity is nearly uniform along the free surface behind the stern. In Figure 13, the instantaneous vertical velocity for the powered case is shown from the same location and on the same scale. The swirl in the velocity field generated by the propulsors are clearly seen and indicates that the propulsors are rotating outboard over the top. The velocities are much more extreme at the free surface behind the stern, which results in the drop in the free surface elevation along the stern.

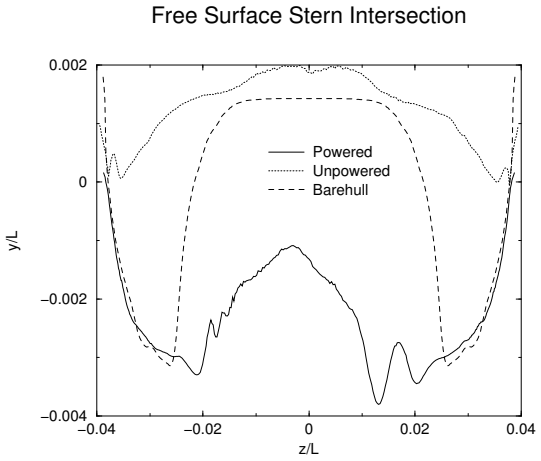

Figure 11: Free Surface / Stern Intersection

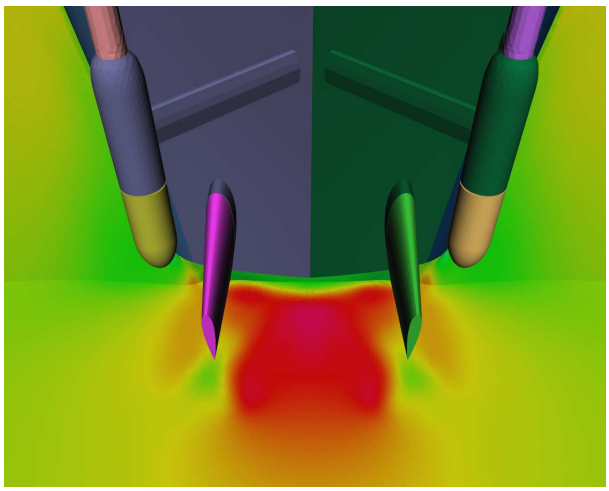

Figure 12: Vertical Velocity for the Unpowered 5415.

POWERED FULLY-APPENDED 5415

The powered fully-appended 5415 simulation was performed using the same parameters as the unpowered case, with the exception that the powered case rotated the actual propulsors used in the experiment. As a result, a periodic solution is expected for the powered case. Figure 14 shows the drag on the port propulsor as well as the total drag on the hull, struts, shafts and rudders and on the

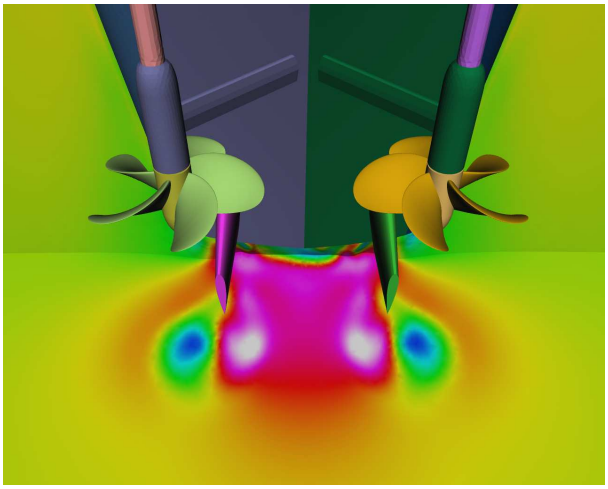

Figure 13: Vertical Velocity for the Powered 5415 (Instantaneous).

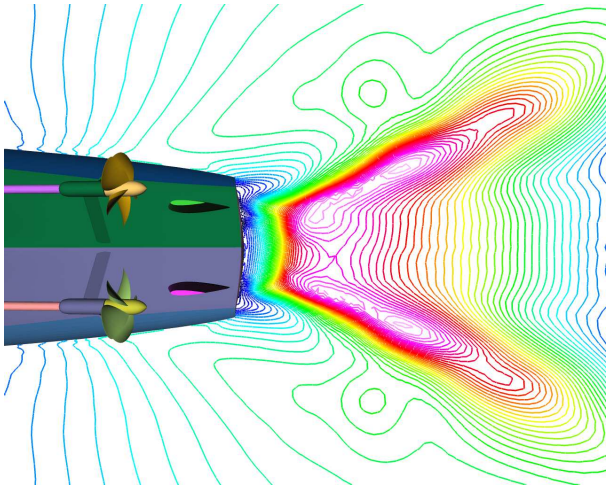

Figure 15: Contours of Free Surface for Stern Region of Powered 5415

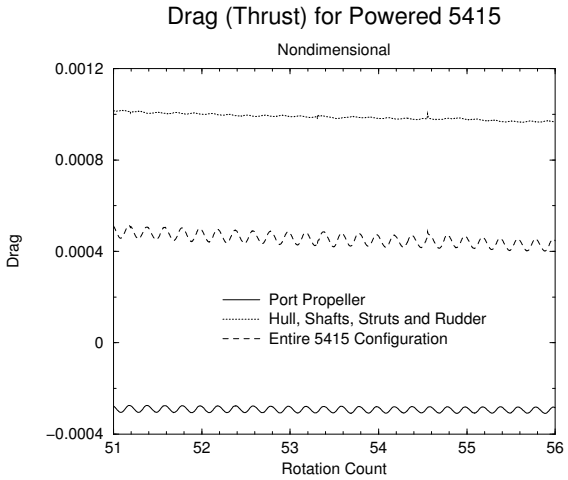

Figure 14: Drag (Thrust) on Powered 5415.

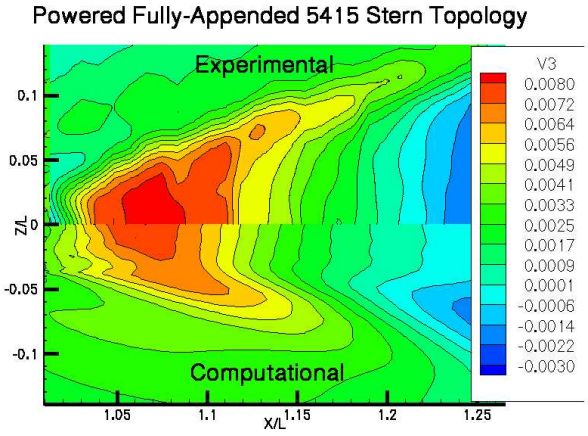

Figure 16: Comparison with Experimental Data for Powered 5415

overall 5415. The drag on the propulsors is negative because the propulsors are introducing thrust into the simulation. The drag is clearly periodic, approximately repeating every 1/5 of the propulsor rotation, since the propulsors had 5 blades. The overall drag is reducing slightly indicating that a highly converged solution has not yet been achieved.

A contour plot of the free surface in the stern region for the powered case is shown in Figure 15 and shows that the flow field is nearly symmetric. Finally, the stern region for the powered case is compared against the experimental results from NSWCCD in Figure 16. The experimental results are time-averaged, whereas the computational results are instantaneous. The height of the "rooster tail" for the computational results is higher than for the experimental results. This difference may be caused by the assumption in the nonlinear free surface algorithm that the free surface is steady-state when the flow is clearly unsteady.

CONCLUSIONS

A robust, nonlinear free surface algorithm coupled with an incompressible Navier-Stokes solver on three-dimensional unstructured grids has been presented. The underlying unstructured solver has been developed to simulate turbulent boundary layer effects and is capable of performing rudder and propulsor induced maneuvering. The addition of nonlinear free surface capabilities is a necessary step towards the simulation of maneuvering surface vessels.

The nonlinear free surface algorithm is an extension of the algorithm used in the structured code UN-

CLE developed by researchers at Mississippi State University. The kinematic free surface equation is solved at each time level, and the hydrostatic pressure is imposed on the Navier-Stokes solver via a characteristic variable boundary condition. After several iterations, the grid is moved to match the free surface while conforming to the underlying geometry, using a three dimensional extension of the torsional spring method (Farhat, 1998). By allowing the grid to move, a nonlinear free surface solution is obtained.

In this paper, the free surface around the fully-appended DTMB Model 5415 series hull with and without rotating propulsors has been simulated. The hull profiles for both simulations upstream of the propulsors is in close agreement with each other and adequate agreement with the experimental data and with computational results using the structured solver UNCLE. The contours of the free surface elevations for both simulations show that the free surface simulation is symmetric even though the grid was not symmetric. And the flow in the stern region for both geometries is symmetric and agrees well with the experimental data generated by Ratcliffe at NSWCCD. Differences in the free surface at the stern can be attributed to differences in the velocity field between the propulsors and between the rudders. More work is necessary in demonstrating that the powered simulation does have a periodic free surface, while the current work has shown that the drag on the geometry is nearly periodic.

Future work includes performing these same simulations at full scale Reynolds number, analyzing the effects of stern flaps on the Model 5415 series hull forms and Series 60 $C_B = 0.6$ ships, simulating flow around unconventional hull forms and performing prescribed and propulsor/rudder induced maneuvers.

ACKNOWLEDGMENTS

This research was sponsored by the Office of Naval Research with Dr. L. Patrick Purtell grant monitor. This support is gratefully acknowledged. Special thanks to Toby Ratcliffe at Naval Surface Warfare Center Carderock Division for providing the data files for the experimental data and to Stephen Nichols at the Computational Simulation and Design Center for many hours of conversations about free surface issues, especially pertaining to the structured code UNCLE.

References

[1] Anderson, W. K., and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," Computers in Fluids, Vol. 23, No. 1, Jan. 1994, pp. 1-21.

[2] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," Proceedings of the 35th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 97-0643, Jan. 1997.

[3] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," 27th AIAA Aerospace Sciences Meeting AIAA Paper 89-0115, Jan. 1989.

[4] Beam, R. M., and Warming, R. F., "An implicit factored scheme for the compressible Navier-Stokes equations," AIAA Journal, Vol. 16. No. 4, April 1978, pp. 393-402.

[5] Beddhu, M., Taylor, L. K., and Whitfield, D. L., "A Time Accurate Calculation Procedure for Flows with a Free Surface Using a Modified Artificial Compressibility Formulation," Applied Mathematics and Computation, Vol. 65, 1994, pp. 33-48.

[6] Beddhu, M., Jiang, M. Y., Taylor, L. K., and Whitfield, D. L., "Computation of Steady and Unsteady Flows with a Free Surface Around the Wigley Hull," Applied Mathematics and Computation, Vol. 89, 1998, pp. 67-84.

[7] Beddhue, M., Jiang, M. Y., Whitfield, D. L., Taylor, L., K., and Arabshahi, A., "CFD Validation of the Free Surface Flow Around DTMB Model 5415 Using Reynolds Averaged Navier-Stokes Equations," Proceedings of the Third Osaka Colloquium on Advanced CFD applications to Ship Flow and Hull Form Design, 1998.

[8] Beddhu, M., Nichols, S., Jiang, M. Y., Sheng, C., Whitfield, D. L., and Taylor, L. K., "Comparison of EFD and CFD Results of the Free Surface Flow Field about the Series 60 C_B = 0.6 Ship," Proceedings of the Twenty-Fifth American Towing Tank Conference, 1998.

[9] Beddhu, M., Jiang, J. Y., Whitfield, D. L., and Taylor, L. K., "Computation of the Wetted Transom Stern Flow over Model 5415," Proceedings of the Seventh International Conference on Numerical Ship Hydrodynamics 1999.

[10] Beddhu, M., Pankajakshan, R., Jiang, M. Y., Taylor, L., Remotigue, M. G., Briley, W. R., and Whitfield, D. L., "Computation and Evaluation of CFD Results for Practical Ship Hull Forms," Proceedings of Gothenburg 2000: A Workshop on Numerical Ship Hydrodynamics, 2000.

[11] Burg, C. O. E., Sreenivas, K., Hyams, D., and Mitchell, B., "Unstructured Nonlinear Free Surface Solutions: Validation and Verification" Proceedings of the 32nd AIAA Fluid Dynamic Conference and Exhibit, AIAA Paper 02-2977, June 2002.

[12] Chorin, A. J., "A numerical method for solving incompressible viscous flow problems," Journal of Computational Physics, Vol. 2, 1967, pp. 12–26.

[13] Coakley, T. J., Hsieh, T., "A comparison between implicit and hybrid methods for the calculation of steady and unsteady inlet flows." Proceedings of the 21st AIAA/SAE/ASME/ASEE Joint Propulsion Conference, AIAA Paper 85-1125, July 1985.

[14] Farhat, C., Degand, C., Koobus, B., and Lesionne, M., "Torsional Springs for Two-Dimensional Unstructured Fluid Meshes," Computer methods in Applied Mechanics and Engining, Vol. 163, 1998, pp. 231-245.

[15] Haselbacher, A., McGuirk, J. J., and Page, G. J., "Finite volume discretization aspects for viscous flows on mixed unstructured grids," AIAA Journal, Vol. 37, No. 2, Feb. 1999, pp. 177–184.

[16] Hyams, D. G., Sreenivas, K., Sheng, C., Briley, W. R., Marcum, D. L., and Whitfield, D. L. "An investigation of parallel implicit solution algorithms for incompressible flows on multielement unstructured topologies," Proceedings of the 38th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2000-0271, January 2000.

[17] Hyams, D. G., "An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Unstructured Topologies," PhD thesis, Mississippi State University, May 2000.

[18] Hyams, D. G., Sreenivas, K., Sheng, C., Nichols, S., Taylor, L. K., Briely, W. R., and Whitfield, D. L., "An Unstructured Multielement Solution Algorithm for Complex Geometry Hydrodynamic Simulations," Proceedings of the Twenty-Third Symposium on Naval Hydrodynamics, September, 2000.

[19] Janus, J. M. " Advanced 3-D CFD Algorithm for Turbomachinery," PhD thesis, Mississippi State University, May 1989.

[20] Marcum., D. L. Private communications, Engineering Research Center for Computational Field Simulation, Mississippi State, MS, December 1997.

[21] Marcum, D. L. "Unstructured Grid Generation Using Automatic Point Insertion and Local Reconnection," The Handbook of Grid Generation, edited by J.F. Thompson, B. Soni and N.P. Weatherill, CRC Press, Boca Raton, 1998, Section 18.

[22] Ratcliffe, T. J., and Lindenmuth, W. T., "Kelvin Wake Measurements Obtained for Five Surface Ship Models," DTRC-89/038, January, 1990, David Taylor Model Basin.

[23] Ratcliffe, T. J., "An Experimental and Computational Study of the Effects of Propulsion on the Free-Surface Flow Astern of Model 5415," Proceedings of the Twenty-third Symposium on Naval Hydrodynamics, September 2000.

[24] Roe., P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," Journal of Computational Physics, Vol. 43, pp. 357–372, 1981.

[25] Sheng, C. Hyams, D., Sreenivas, K., Gaither, A., Marcu, D., Whitfield, D., and Anderson, W. K., "Three-dimensional incompressible Navier-Stokes flow computations about complete configurations using a multiblock unstructured grid approach," Proceedings of the 37th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 99-0778, January 1999.

[26] Singh, K. P., Newman, J. C., and Baysal, O., "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion," AIAA Journal, Vol. 33, No. 4, April 1995, pp. 641-649.

[27] Spalart, P. R., and Allmaras, S. R., "A one-equation turbulence model for aerodynamic flows," AIAA Paper 92-0439, 1992.

[28] Taylor, L., K., "Unsteady Three-Dimensional Incompressible Algorithm based on Artificial Compressibility," PhD thesis, Mississippi State University, 1991.

[29] Taylor, L. K., Busby, J. A., Jiang, M. Y., Arabshahi, A., Sreenivas, K., and Whitfield, D. L., "Time accurate incompressible Navier-Stokes simulation of the flapping foil experiment." Proceedings of the Sixth International Conference on Numerical Ship Hydrodynamics, August 1993.

[30] Thomas, P. D., and Lombard, C. K., "Geometric conservation law and its application to flow computations on moving grids" AIAA Journal, Vol. 17, No. 10, 1978, pp. 1030–1037.