AIAA 2009-3887

Hybrid Finite Element and Finite Volume Methods for Two Immiscible Fluid Flows

Shahrouz Aliabadi[†], Tian Wan[†], and Christopher Bigler[‡]

[†] Jackson State University, School of Engineering
Northrop Grumman Center for High Performance Computing, Jackson, MS

[‡] University of Michigan, Department of Naval Architecture and Marine Engineering
Ann Arbor, MI

We have successfully extended our implicit hybrid finite element/volume solver to flows involving two immiscible fluids. The solver is based on the segregated pressure correction or projection method on staggered unstructured hybrid meshes. An intermediate velocity field is first obtained by solving the momentum equations with the matrix-free implicit cell-centered finite volume method. The pressure Poisson equation is solved by the node-based Galerkin finite element method for an auxiliary variable. The auxiliary variable is used to update the velocity field and the pressure field. The pressure field is carefully updated by taking into account the velocity divergence field. This updating strategy can be rigorously proven to be able to eliminate the unphysical pressure boundary layer and is crucial for the correct temporal convergence rate. Our current staggered-mesh scheme is distinct from other conventional ones in that we store the velocity components at cell centers and the auxiliary variable at vertices. The fluid-interface is captured by solving an advection equation for the volume fraction of one of the fluids. The same matrix-free finite volume method as the one used for momentum equations is used to solve the advection equation. We will focus on the interface sharpening strategy to minimize the smearing of the interface over time. We have developed and implemented a global mass conservation algorithm which enforces the conservation of the mass for each fluid.

I. Introduction

Recently, we developed a hybrid finite element/volume (FE/FV) solver [1] for incompressible flows. The hybrid solver is based on the well-known pressure correction (projection) method [2, 3]. The solution procedure follows a segregated approach to decouple the pressure from the velocity. The velocity field is updated by solving the momentum equation provided that a known pressure field is given as a source term, through a cell-centered finite volume (FV) discretization. The pressure does not directly enter the momentum equation. Instead, an auxiliary variable, which is closely related to the pressure, takes the place of pressure in the momentum equation, providing pressure gradient information. We put the auxiliary variable on the vertices of cells. This deployment provides a convenient way to evaluate the pressure gradient using the local finite element basis functions. The incremental value of the auxiliary variable is computed by solving a Poisson equation using the Galerkin finite element (FE) method. The auxiliary variable is then used to update the velocity field. After the final velocity field is determined, the pressure can be updated using the auxiliary variable and the velocity divergence field. The pressure is updated in such a way that the pressure field is free of unphysical conditions in the boundary layer.

Our hybrid finite volume/element solver is aimed to take advantage of the merits of both the FV and the FE methods and avoid their shortcomings. For example, highly-stretched cells (also known as high-aspect-ratio cells) are commonly used inside the boundary layer for high Reynolds number flows to resolve the boundary layer and reduce the number of cells. The stabilization parameters in the stabilized FE based flow solvers [4, 5] are related to the characteristic element length that is not well defined for high-aspect-ratio mesh elements. Due to this, it is very difficult to control the numerical dissipation of stabilized finite element solvers. By contrast, the finite volume flow solver is very insensitive to the aspect ratio of the mesh cells. It is quite common for the FV solvers to handle cells with aspect ratios in the order of thousands [6, 7]. For this reason, we use the finite volume method to solve the momentum equation. On the other hand, the classic Galerkin FE method is very suitable for the elliptic typed equations like the pressure Poisson equation emerging from the segregated approach. Therefore, the combination of the FV method and the FE method is expected to perform well in the incompressible flow solvers based on the pressure projection method, which has been confirmed by our earlier work [1]. The numerical examples we presented in [1] are all about low Reynolds number flows. In our recent work, we have extended the hybrid flow solver to high Reynolds number flows using hybrid meshes with high aspect ratios [8].

We also incorporated the Detached Eddy Simulation (DES) [9] turbulence model into the flow solver to compute the eddy viscosity. The DES model was originally proposed to be an affordable hybrid Reynolds-averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES) models for flows at realistic Reynolds numbers. In attached boundary layers the DES model acts as a RANS model and in massively separated regions the DES functions as the LES model. A modified distance to the wall named the DES distance acts as a switch between the RANS mode and the LES mode [9]. There are two DES-based turbulence models. One is the one-equation Spalart-Allmaras (SA) DES model [9-10]. The other is the two-equation Shear Stress Transport (SST) DES model [11]. The SA-DES model is gaining more popularity [12-14] due to its simplicity and fair accuracy. In our hybrid flow solver, we incorporated the SA-DES turbulence model. For details of implementations, see [8].

Generally, there are two distinct approaches in the numerical simulation of two-fluid flows (excluding panel methods [15]). Depending on the physical characteristics of the problem, either “moving-mesh” or “fixed-mesh” techniques are used. In the moving-mesh techniques, the motion of the free-surface is absorbed by moving the computational nodes located on the free-surface [16-17]. Most of the moving-mesh techniques are based on either the space-time finite element formulations [18-19] or the Arbitrary Lagrangian-Eulerian (ALE) [20-21] formulations. In the applications where the deformation of the free-surface is large, the moving-mesh techniques usually result in element distortions. As the element distortions grow and become unacceptable, the generation of a new mesh and the projection of the solution from the old mesh to the new one is essential [22]. In complex 3D applications, this procedure is extremely difficult and time consuming. In such cases, computations using fixed-mesh techniques are more desirable.

The most common fixed-mesh techniques are based on the VOF [23], the level-set [24-25] and Interface-Sharpening/Global Mass Conservation (IS-GMC) methods [26-27]. In these methods, the Navier-Stokes equations are solved over a non-moving mesh. A scalar function (or color function) acts as a marker to identify the location of the free-surface. This function is transported throughout the computational domain with a transient advection equation.

In this article we will describe the extension our implicit hybrid finite element/volume solver to flows involving two immiscible fluids based on IS-GMC algorithm. In our implementation, we will focus on the free surface sharpening strategy to minimize the smearing of the interface over time.

The rest of this paper is organized as follows. Section 2 presents the governing equations. Section 3 provides the solution strategy and a brief overview of our hybrid solver. Section 4 provides some implementation issues regarding the mass conservation. Section 5 deals with parallel implementation and parallel performance. Some test cases are presented in Section 6 to demonstrate the performance of the current two fluid-flow solver. Finally in Section 7, we summarize this paper with final concluding remarks.

II. GOVERNING EQUATIONS

The velocity and pressure fields of two isothermal incompressible and immiscible fluids, denoted by Fluid^A and Fluid^B, moving in the domain $\mathbf{x} \in \Omega$ with boundary $\partial\Omega$ during the time interval $t \in (0,\mathrm{T})$ can be described by the incompressible Navier Stokes equations for multiphase flows. Let $\Omega = \Omega_t^A \cap \Omega_t^B$ where Ω_t^A and Ω_t^B represent volumes occupied by Fluid^A and Fluid^B at time t, respectively, and ϕ denotes the volume of the fluid function such that :

$$\phi = \begin{cases} 1 \text{ on } \Omega_t^A \\ 0 \text{ on } \Omega_t^B \end{cases} \tag{1}$$

Using ϕ, we can define the density, ρ, and the viscosity, μ, as following:

$$\rho = \rho_A + \left(\rho_B - \rho_A\right)\phi \tag{2}$$

$$\mu = \mu_A + \left(\mu_B - \mu_A\right)\phi \tag{3}$$

where the subscription “A” and “B” refer to Fluid^A and Fluid^B respectively. For these two fluids, the Navier-Stokes equations can be written as:

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{5}$$

$$\rho\left[\frac{\partial \mathbf{u}}{\partial \mathrm{t}} + \left(\mathbf{u} \cdot \nabla\right)\mathbf{u}\right] = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}\,; \qquad \boldsymbol{\sigma} = - p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \tag{6}$$

Here, $\mathbf{u}$, p, and $\mathbf{g}$, are the fluid velocity, pressure and gravitational force, respectively. Eqs. (4-6) are completed by an appropriate set of boundary and initial conditions. Based on the velocity, the entire physical boundary can be generally categorized into $\partial\Omega = \Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{n}} \cap \Gamma_{\mathrm{N}}$ (the notation follows that in [28]) where

- Γ_{D} : $\mathbf{u} = \mathbf{u}_{\mathrm{b}}$, i.e. all components of velocity are specified, e.g., inflow boundary.
- Γ_{n} : $\mathbf{u} \cdot \mathbf{n} = (\mathbf{u} \cdot \mathbf{n})_{\mathrm{b}}$, i.e. only the velocity component normal to the boundary is specified, e.g., symmetry boundary, and
- Γ_{N} : $\boldsymbol{\sigma} \cdot \mathbf{n} = (\boldsymbol{\sigma} \cdot \mathbf{n})_{\mathrm{b}}$, e.g., outflow boundary.

where $\mathbf{n}$ is the unit vector normal to the boundary. In addition to physical boundary conditions, at the moving interface of two fluids, the jump condition applies. For two immiscible fluids we can write:

$$\begin{cases} \mathbf{n} \cdot \left(\boldsymbol{\sigma} \cdot \mathbf{n}\right) = \gamma\kappa \\ \mathbf{s} \cdot \left(\boldsymbol{\sigma} \cdot \mathbf{n}\right) = 0 \end{cases} \tag{7}$$

where $\mathbf{s}$ is the unit vector tangent to the interface. Here γ and κ are the coefficient of surface tension and radius of the curvature, respectively. Using Eq. (7), one can obtain the following equivalent equation across the interface [29]:

$$p_A\, \mathbf{n} = p_B\, \mathbf{n} + \gamma\kappa\, \mathbf{n} \tag{8}$$

The momentum equation, Eq. (6), is often referred to as the stress-divergence form [28] which is the dominantly adopted form in finite element based flow solvers [5, 26-27] due to the convenience of enforcing the natural boundary conditions in the weak form of Eq. (6). Alternatively, the incompressible momentum equations can also be written as

$$\rho\left[\frac{\partial \mathbf{u}}{\partial t} + \left(\mathbf{u} \cdot \nabla\right)\mathbf{u}\right] = \rho \mathbf{g} - \nabla p + \nabla^2 \mathbf{u} \tag{9}$$

which is the simplified form by taking into account $\nabla \cdot \mathbf{u} = 0$ in Eq. (6). The momentum equation in the form of Eq. (9) is more commonly used in non-finite element based flow solvers [2, 3]. Our experience shows that Eq. (6) and Eq. (9) yield almost identical results using our current solver. However, the outflow boundary condition should be handled properly to account for the different forms of the stress tensor.

We use one-equation Spalart-Allmaras DES model [9] which is derived from the Spalart-Allmaras one-equation eddy-viscosity RANS model. The equation can be written as:

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\nu} = \frac{1}{\Psi \mathrm{Re}}\left[\nabla \cdot \left(\left(\nu + \tilde{\nu}\right)\nabla \tilde{\nu}\right) + c_{b2}\left|\nabla \tilde{\nu}\right|^2\right] + c_{b1}\tilde{S}\tilde{\nu} - \frac{c_{w1} f_w}{\mathrm{Re}}\left(\frac{\tilde{\nu}}{d_{DES}}\right)^2 \tag{10}$$

where $\tilde{\nu}$ is the working variable in the order of the molecular viscosity ν and Re is the Reynolds number. If the molecular viscosity is constant, then $\nu = 1$. In Eq. (10), the second and the last term on the right hand side are the production term and destruction term, respectively. For details see [8].

III. Solution Strategy

Our hybrid incompressible flow solver is based on the pressure correction or projection method. A fractional step approach is employed to decouple the pressure from the velocity. A truly matrix-free [7, 30] (both the GMRES and preconditioning are matrix-free) implicit cell-centered finite volume method is used to solve the original momentum equations to obtain an intermediate velocity field. The node-based Galerkin finite element method is used to solve the Poisson equation derived from the fractional step approach for an auxiliary variable that is closely related to the real pressure. The auxiliary variable is used to update the velocity field and the pressure field. We store the velocity components at cell centers and the auxiliary variable at cell vertices, making the current hybrid solver a staggered-mesh scheme which is, however, distinct from other conventional staggered grid schemes where velocity components are usually stored at cell edges and the pressure at cell centers.

At the beginning of each time-step, we are given an initial velocity field $\mathbf{u}^n$ (not necessarily divergence-free), ϕ^n and $\tilde{\nu}^n$ at cell centers and auxiliary variable field q^n at vertices, where the superscript 'n' stands for the previous time-step. At the end of the time-step, $\mathbf{u}^n$, ϕ^n, $\tilde{\nu}^n$ and q^n will be updated to $\mathbf{u}^{n+1}$, ϕ^{n+1}, $\tilde{\nu}^{n+1}$ and q^{n+1}, respectively. At the same time the real pressure, p, will be updated as a by-product. To obtain $\mathbf{u}^{n+1}$, ϕ^{n+1}, $\tilde{\nu}^{n+1}$ and q^{n+1}, we perform the following

- Step 1: Set $\mathbf{u} = \mathbf{u}^n$ $\phi = \phi^n$, $\tilde{\nu} = \tilde{\nu}^n$ and $q = q^n$
- Step 2: Solve the equation governing the motion of the interface function (Eq. (11)) to update value of ϕ.

$$\frac{\alpha_1 \phi + \alpha_0 \phi^n + \alpha_{-1} \phi^{n-1}}{\Delta t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{11}$$

Note that in Eq. (11), the time-dependent term has been discretized using the backward difference formula (BDF). For first order time accurate scheme (BDF1), $\alpha_1 = 1.0$, $\alpha_0 = -1.0$ and $\alpha_{-1} = 0.0$ and for second order time accurate scheme (BDF2), $\alpha_1 = 1.5$, $\alpha_0 = -2.0$ and $\alpha_{-1} = 0.5$.

Equation (11) is solved by the implicit cell-centered finite volume method. The current FV solver is truly matrix free. The Jacobian-free version of the Generalized Minimal RESidual method (GMRES) solver is implemented to solve the resulting linear system after implicit discretization. Moreover, the matrix-free Lower-Upper Symmetric Gauss Seidel (LU-SGS) preconditioner is adopted to accelerate the GMRES convergence. More details can be found in [1].

- Step 3: Similarly, using matrix free FV method, solve the turbulent equation discretized in the BDF scheme (Eq. (12)).

$$\frac{\alpha_1 \tilde{\nu} + \alpha_0 \tilde{\nu}^n + \alpha_{-1} \tilde{\nu}^{n-1}}{\Delta t} + \mathbf{u} \cdot \nabla \tilde{\nu} = \frac{1}{\Psi \mathrm{Re}} \left[\nabla \cdot \left(\left(\nu + \tilde{\nu} \right) \nabla \tilde{\nu} \right) + c_{b2} \left| \nabla \tilde{\nu} \right|^2 \right] + c_{b1} \tilde{S} \tilde{\nu} - \frac{c_{w1} f_w}{\mathrm{Re}} \left(\frac{\tilde{\nu}}{d_{DES}} \right)^2 \tag{12}$$

Refer to [8] for implementation details and robust approach to solve the Eq. (12).

- Step 4: Using the same matrix free FV method, solve the time-discretized momentum equation (Eq. (13)) to obtain the intermediate velocity field $\tilde{\mathbf{u}}$.

$$\rho \frac{\alpha_1 \tilde{\mathbf{u}} + \alpha_0 \mathbf{u}^n + \alpha_{-1} \mathbf{u}^{n-1}}{\Delta t} + \nabla q^n = \rho (\mathbf{g} - \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}) + \mu \nabla^2 \tilde{\mathbf{u}} \tag{13}$$

It should also be noted that it is the auxiliary variable q instead of the real pressure p that enters the momentum equation.

- Step 5: Solve the pressure correction equation to obtain the incremental auxiliary variable q' using Eq. (14).

$$\nabla^2 q' = \frac{\rho \alpha_1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}} \tag{14}$$

Eq. (14) is a Poisson equation (elliptic) which is solved by the continuous Galerkin FE method. Since the stiffness matrix is symmetric after the FE discretization, we are able to use the preconditioned Conjugate Gradient (CG) method to solve the system. The preconditioner is based on the symmetric Gauss-Seidel (SGS) decomposition of the stiffness matrix which is also an incomplete LU decomposition (ILU(0)). The Eisenstat trick [31] is utilized in the implementation. With the Eisenstat trick, the preconditioner adds negligible extra cost compared with non-preconditioned solver.

- Step 6: Update the auxiliary variable q. The auxiliary variable is updated according to

$$q = q^n + q'\,. \tag{15}$$

- Step 7: Update the velocity $\mathbf{u}$. The velocity is updated according to

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho \alpha_1} \nabla q' . \tag{16}$$

Since the incremental auxiliary variable q' is stored at the vertices of the mesh and the velocity vector is stored at the cell centers, the gradient of q' can be computed via the local elemental FE basis function evaluated at the cell center.

- Step 8: In the Newton-Raphson nonlinear iteration, update and repeat Steps 1-7 until desirable convergence tolerance is obtained.
- Step 9: Update $\mathbf{u}^{n+1} = \mathbf{u}$, $\phi^{n+1} = \phi$, $\tilde{\nu}^{n+1} = \tilde{\nu}$ and $q^{n+1} = q$
- Step 10. Update the real pressure p^{n+1}. The real pressure is updated according to:

$$p^{n+1} = q^{n+1} - c\mu \nabla \cdot \tilde{\mathbf{u}} . \tag{17}$$

where $c = 2$ if Eq. (6) is used and $c = 1$ if Eq. (9) is used.

Note that in the current hybrid solver, the real pressure appears as a by-product in the solution procedure. The real pressure is updated utilizing the auxiliary variable field and the velocity-divergence field. The velocity-divergence contains the velocity gradient information which has been obtained when solving the momentum equations. This pressure updating strategy leads to the correct physical pressure Neumann boundary condition (see [1] for detailed proof) on Γ_D and Γ_n.

As can be seen from the aforementioned ten steps, the solution procedure is very efficient. Step 5, solving a Poisson equation, is the most computational intensive step (approximately 60% of the computational cost).

IV. INTERFACE SHARPENING/GLOBAL MASS CONSERVATION (IS-GMC)

In Step 2, we solve purely a hyperbolic system to advance the front of the interface function with the fluid velocity. Since the interface function is a discontinuous function, its accurate representation over discrete domain is a major challenge. In fact, most of the algorithms developed for free-surface or two immiscible fluid flows suffer from global mass conservation. The error associated in solving the hyperbolic equation governing the advance of interface front results is fictitious mass transfer between the two fluids.

The numerical solution of Eq. (11) will result in a diffusive solution with over-shoots and undershoots in the vicinity of the interface. Since Eq. (11) is solved using our advanced finite volume method [32], a slope limiting procedure (described in [32]) will eliminate the overshoots and undershoots. However, the diffusive interface still will introduce inaccuracies in mass conservation. To recapture the mass conservation, we sharpen the diffusive interface ϕ with ϕ_{new} as following:

$$\phi_{\text{new}} = \begin{cases} \beta^{1-\alpha}\phi^{\alpha} & 0 \le \phi \le \beta \\ 1\text{-}(1\text{-}\beta)^{1-\alpha}(1\text{-}\phi)^{\alpha} & \beta \le \phi \le 1 \end{cases} \tag{18}$$

where $1 \le \alpha$ is a sharpening parameter [26-27], and $0 \le \beta \le 1$ (unknown) is a point satisfying the global conservation of mass for each fluid. To determine β, we satisfy the mass conservation over entire computational domain at a given time *t* for ϕ_{new}. Note that the total mass of each fluid is the summation of the original mass plus the net amount of mass flux entering or exiting through the physical boundaries. Therefore:

$$\rho_A \int_{\Omega} \phi_{\text{new}} \, d\Omega = m_A + \rho_A \int_t \int_{\Gamma} \phi \, \mathbf{u} \cdot \mathbf{n} \, d\Gamma dt \tag{19}$$

$$\rho_B \int_{\Omega} (1 - \phi_{new}) d\Omega = m_B + \rho_B \int_t \int_{\Gamma} (1 - \phi) \, \mathbf{u} \cdot \mathbf{n} \, d\Gamma dt \tag{20}$$

where m_A and m_B are the initial mass of Fluid^A and Fluid^B, respectively. Note that we only need to satisfy either Eq. (19) or Eq. (20). Inserting Eq. (18) into either equation assuming that the parameter α is given and constant, we can obtain:

$$M\beta^{(1\text{-}\alpha)} + N(1\text{-}\beta)^{(1\text{-}\alpha)} = K \tag{21}$$

where M, N, K are all functions of β. This nonlinear equation is solved using a Newton-Raphson algorithm. Typically, with the initial guess of 0.5, the algorithm converges in two or three nonlinear iterations.

Once the parameter β is obtained, ϕ will be replaced by ϕ_{new}. It is worthy to mention that the parameter β will define where the true interface is at each time-step.

V. Parallel Jacobian-free GMRES Solver

For three-dimensional problems, iterative methods are almost mandatory [36]. The Generalized Minimal RESidual method (GMRES) [36] has been widely used to solve large sparse systems. Because the GMRES algorithm involves only matrix-vector multiplication, it is unnecessary to form the Jacobian explicitly. The details of forming the Jacobian-vector production can be found in our previous paper [7]. The Jacobian-free GMRES solver has the following advantages:

- Avoid the difficulty and cost in forming the Jacobian matrix. For high order finiter volume solvers, the analytic evaluation of the Jacobian matrix becomes difficult, if not impossible.
- Save a lot of memory for storing the Jacobian matrix, which may become the bottleneck in order to solve multiple degree-of-freedom Partial Differential Equations on billion-element mesh.

Of course, the disadvantage of not forming the Jacobian matrix is obvious: we have to evaluate the Jacobian-vector production many times depending on the size of the Krylov space.

The choice of preconditioner is crucial for the convergence of GMRES solver. At the present stage of this paper, the following preconditioners are implemented:

- For the momentum, turbulence and free-surface equation, the matrix-free LU-SGS preconditioner [37] is used, which forms the preconditioning matrix with a lower order flux scheme. It's easy to parallelize and gives good convergence. In the final version of the paper, we will show the convergence comparison of LU-SGS versus diagonal preconditioner.
- For the pressure equation, either diagonal or ILU preconditioner is used. The diagonal preconditioner is easy to parallelize and matrix-free, whereas ILU requires matrix storage and special attention taken for parallelization. In the final paper, we will show the convergence comparison of both preconditioner.

The present solver has been parallelized on clusters using the ParMETIS mesh partitioning [38] and the MPI parallel programming module. Due to the data structures we use, we have communications requirement for nodes, faces and cells. Very efficient non-blocking MPI functions are called to set up the inter-processor "gather" and "scatter" routines in the pre-processing stage.

Scaling test has been taken on an eight-core cluster. The cluster has ten nodes each containing two quad-core Intel Xeon processors at 3.0 Ghz. Each node has 32 GB memory, and the nodes are connected by Infini-band. Intel Fortran and C compilers are used, together with OpenMPI. OpenMPI is fully compatible with Infini-band, and as a result the communcation latency of our code is very low. The preliminary scaling test is performed on a mesh of 1.3 million elements and 1.2 million nodes. In the final paper, a much refined mesh will be used. Figure 1 shows the test results on 8, 16, 32 and 64 processors. The CPU time of 8 processors is used as the baseline to compute speedup, and for the purpose of comparison, in the figure its speedup is plotted as 8. The results show perfect linear speedup up to 32 processors; for the case of 64 processors the speedup is lower down, since the communcation/computation CPU time ratio increases as the number of processor increases, and at this point the computation is not long enough to hide the non-block communication. In the final version of the paper, we will do the scaling test on a much finer grid and show the detailed break-down of the CPU time on communication and solving each equation.

VI. Numerical Examples

Three standard test problems for free-surface flows are chosen to validate our hybrid two-fluid flow solver. In all three problems, the non-dimensional properties of water and air are set as:

$$\frac{\rho_{\text{water}}}{\rho_{\text{air}}} = 813, \quad \frac{\mu_{\text{water}}}{\mu_{\text{air}}} = 55 \tag{22}$$

Breaking Dam: The first test problem is the breaking dam problem [33], which is a classic test case for free surface flows. In this problem, a column of the water is suddenly released and fall into ground due to gravitational force. The computational domain and the problem definition are shown in Figure 2.

We used two structured meshes consisting of uniform quadrilateral elements to simulate this problem. The coarse and refined meshes are discretized with 125x100 and 260x210 elements in the horizontal and the vertical directions, respectively. Figure 3 shows the simulation results by plotting the volume of the fluid variable at four instances. The images in Figure 3 show that the interface between air and water is captured very well by our solver, and the transition region of the interface is very small in size. In Figure 4, the horizontal displacement of the interface front on the ground is compared with experimental data [33]. The agreement is very good for both coarse and refined meshes. In Figure 4, the dimensionless time and displacement are given by $\tau = t\sqrt{2g / a}$ and $\delta = x / a$, where a is the initial width of the water column.

Oscillating Flow in a Tank. In our second test problem, the oscillating flow in a tank is studied. This problem further allows us to evaluate the capability of our method to capture interface dynamics (also refereed as free-surface) in air-water interaction. Our computed results are compared with those obtained by Iafrati et al. in [34]. The tank extends horizontally from $x = -0.5$ to $x = 0.5$ and from $y = 0$ to $y = 1.4$ in the vertical direction. The computational domain is non-dimensionalized with horizontal length. The initial free surface is described by the equation

$$y(x) = 1 - 0.01\sin(\pi x) \tag{23}$$

From the equation it is clear that initially the y value of the interface on the left wall is 1.01 and on the right wall 0.99. For this flow, $U = \sqrt{gL}$ is used as reference velocity. The Froude and Reynolds numbers are set to one and one million, respectively. The structured mesh for this test case has 40 equally spaced points in the horizontal direction and 280 points in the vertical direction, suitably clustered close to the interface region. In the region $0.985 \le y \le 1.015$, a uniform vertical spacing of $\Delta y = 0.00025$ is adopted.

Figure 5 shows the time history of the free-surface height on the left wall of the tank for our computations and the results obtained in [34]. As we can see, the agreement between the two results can prove further that that the dynamics of the free-surface on the left wall is well captured. Figure 6 plots the velocity vectors around the interface between the air and water. It is clear that the velocity around the interface is smooth without any over-shoot or under-shoot oscillations.

NACA 0012 Submerged Hydrofoil. The last test case is the NACA 0012 submerged hydrofoil at high Reynolds number and an angle of attack of 5°. The quantities are non-dimensionalized with the chord length of the body and the uniform inflow velocity. Based on these scales, the Reynolds number is $\mathrm{Re} = 1.624 \times 10^5$, and the Froude number is $\mathrm{Fr} = 0.5672$. The non-dimensional submergence at mid-chord is 1.034 in our simulation. The computational domain is 18 units in length (the upstream boundary is 6 units upstream of the leading edge) and 6 units in depth.

A hybrid computational mesh with 139,614 elements which incorporates rectangular elements in the boundary layer near the hydrofoil surface and in the vicinity of the interface and triangular elements elsewhere is used in this computation. Figure 7 shows a close-up of the mesh near the hydrofoil.

Due to the high Reynolds number flow in this problem, the transient flow turns turbulent very quickly. Hence the DES turbulence model was turned on in the simulation. As we expected, the unsteady behavior in the flow was observed in our simulations. Figure 8 compares our computed interface profile with the experimental measurements reported in [35].

Figure 9 shows the velocity vectors near the interface between air and water. From the plot, it is clear that the flow is attached to the hydrofoil, as flow is turbulent and separation is prevented. The image in Figure 8 also shows smooth velocity across the interface. Figure 10 depicts the computed pressure at $t = 34\mathrm{s}$, which shows pressure discontinuity across the interface. It is clear that in the water phase, the hydro-pressure increase with water depth due to gravity.

Composite High-Speed Vessel. Northrop Grumman Ship Systems (NGSS) is currently researching the affects of adding a blended wing body system, which incorporates lifting bodies and hydrofoils, to a composite high-speed vessel (CHSV). In modifying the composite monohull to include the blended wing body system, NGSS has included sponsons and aft amas into the concept. The principle dimensions of the vessel are shown in Table 1.

Length Overall	90.60 meters
Length Between Perpendiculars	73.58 meters
Maximum Beam	22.30 meters
Design Draft	3.677 meters
Design Displacement	2,000 metric tons
Maximum Design Speed	40 knots

Table 1. CHSV Principle Dimensions.

The present code is used to perform CFD full-scale predictions and model test predictions of the CHSV at maximum design speed. The computational mesh has around 25.2M elements, and Figure 11 shows the mesh on the symmetry plane. In our final paper, a much refined mesh will be used and the computed results will be compared with those of the present mesh.

Analysis was completed at a 3.677m static draft. Again, the speed of flow is 40 knots. The vessel displacement is 2000 metric tons, yielding a ship weight of approximately $1.96x10^7$ Newtons. The vertical center of gravity is 4.68 meters above baseline and the longitudinal center of gravity is 27.95 meters forward of the transom. The temperature of the water is 15 degrees Celsius and the water density is 1025 kg/m^3.

The free-surface contours are plotted in Figure 12. The figures show that the wake pattern is captured by our code. Figure 13 plots the pressure distribution on the vehicle. The lift/weight ratio is computed to be 1.19, which means the lift generated by this model is 19% greater than the weight of the vehicle. Figure 14 plots the time history of the dimensionless lift force generated by both pressure and viscous force. It shows that the lift force has reached its steady state.

VII. Conclusion

In this paper, we have successfully extended our implicit hybrid finite element/volume solver to flows involving two immiscible fluids. The solver is based on the segregated pressure correction or projection method on staggered unstructured hybrid meshes. The GMRES matrix-free strategy was adapted to solve the governing equations using finite element and finite volume methods. The fluid-interface is captured by solving an advection equation for the volume fraction of one of the fluids. We implemented interface sharpening strategy to minimize the smearing of the interface over time. We have developed and implemented a global mass conservation algorithm which enforces the conservation of the mass for each fluid. We used LU-SGS preconditioning technique to solve the linear system of equations. Though the LU-SGS preconditioning for the momentum equations and the turbulence equation proved beneficial, a more efficient Poisson equation solver will be sought to improve the efficiency and reduce the overall computational cost of simulations.

Acknowledgments

Work is funded by Northrop Grumman Ship Systems and National Science Foundation.

References

1. Tu S, Aliabadi S. Development of a hybrid finite volume/element solver for incompressible flows on unstructured meshes. International Journal of Numerical Methods in Fluids 2007; 55(2):177–203.
2. Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. Computer Methods in Applied Mechanics and Engineering 2006; 195:6011–6045.
3. Timmermans LJP, Minev PD, van de Vosse FN. An approximate projection scheme for incompressible flow using spectral elements. International Journal of Numerical Methods in Fluids 1996; 22:673–688.
4. Aliabadi S, Johnson AA, Abedi J. Stabilized-finite-element/interface-capturing technique for parallel computation of unsteady flows with interfaces. Computer & Fluids 2003; 32:535–545.
5. Aliabadi S, Tu S, Watts MD, Ji A, Johnson AA. Integrated high performance computational tools for simulations of transport and diffusion of contaminants in urban areas. International Journal of Computational Fluid Dynamics 2006; 20(3–4):253–267.
6. Delanaye M. Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids. Ph.D. thesis, Universite de Liege, 1996.
7. Tu S, Watts MD, Fuller A, Patel R, Aliabadi S. Development and performance of CaMEL_Aero, a truly matrix-free, parallel and vectorized unstructured finite volume solver for compressible flows. Proceedings of the 25th Army Science Conference, Orlando, FL, 2006.
8. Tu S, Aliabadi S, Patel R, Watts M. An implementation of the Spalart-Allmaras DES model in an implicit unstructured hybrid finite volume/element solver for incompressible turbulent flow. To appear in the International Journal of Numerical Methods in Fluids.
9. Spalart PR, Jou W-H, Strelets M, Allmaras SR. Comments on the feasibility of LES for wings and on hybrid RANS/LES approach. Proceedings of the 1st AFOSR International Conference on DNS/LES, Columbus, OH, 1997.
10. Nikitin NV, Nicoud F, Wasistho B, Squires KD, Spalart PR. An approach to wall modeling in large-eddy simulations. Physics of Fluids 2000; 12(7):1629–1632.

11. Strelets M. Detached eddy simulation of massively separated flows. AIAA paper 2001–0879, the 39th AIAA Aerospace Sciences Meeting and Exhibits, Reno, NV, 2001.

12. Scott AM, Matthew BS, Russell MC, James RF. DES grid resolution issues for vortical flows on a Delta wing and an F-18C. AIAA paper 2003–1103, the 41st Aerospace Sciences Meeting & Exhibit, Reno, NV, 2003.

13. Kapadia S, Roy S, Wurtzler K. Detached eddy simulation over a reference Ahmed car model, AIAA paper 2003–0857, the 41st Aerospace Sciences Meeting and Exhibit, Reno, NV, 2003.

14. Forsythe JR, Hoffmann KA, Cummings RM, Squires KD. Detached eddy simulation with compressibility corrections applied to a supersonic axisymmetric base flow. Journal of Fluid Engineering 2002; 124(4):911–923.

15. Rosen BS, Laiosa JP. SPLASH Nonlinear and Unsteady Free-Surface Analysis Code for Grand Prix Yacht Racing. The Thirteenth Chesapeake Sailing Yacht Symposium, Annapolis, MD, Jan. 1997.

16. Aliabadi S, Tezduyar T. Stabilized-Finite-Element/Interface-Capturing Technique for Parallel Computation of Unsteady Flows with Interfaces. Computer Methods in Applied Mechanics and Engineering, 190, 243-261 (2000).

17. Sundel T. Computation of the Free-Surface Flows Around a Ship Using NS Solver FINFLO. VTT Manufacturing Technology, 1997.

18. Aliabadi SK, Tezduyar TE. Space-time Finite Element Computation of Compressible Flows Involving Moving Boundaries and Interfaces. Computer Methods in Applied Mechanics and Engineering, 107, 209-223 (1993).

19. Hughes TJR, Hulbert GM. Space-Time Finite Element Methods for Elastodynamics: Formulations and Error Estimates. Computer Methods in Applied Mechanics and Engineering, 66, 339-363 (1998).

20. Donea J. An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Fluid-Structure Interactions. Computational Mechanics, 33, 689-723 (1982).

21. Farhat C, Lesoinne M, Maman N. Mixed Explicit/Implicit Time Integration of Coupled Aeroelastic Problems: Three-Field Formulation, Geometric Conservation and Distributed Solution. International Journal for the Numerical Methods in Fluids, 21, 807-835 (1995).

22. Johnson A. Tezduyar T. Advanced Mesh Generation and Update Methods for 3D Flow Simulations. Computational Mechanics, 23,130-143 (1999).

23. Hirt W, Nichols BD, “Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. Journal of Computational Physics, 39:201-225 (1981).

24. Sussman M, Smareka P, Osher S. A level Set Approach for Computing Incompressible Two-Phase Flows. Journal of Computational Physics, 114, 146-168, (1994).

25. Sethian JA. Level Set Method. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1996.

26. Aliabadi S, Johnson A., Zellars B, Abatan A, Berger C. Parallel Simulation of Flows in Open Channels. Journal of Future Generation Computer Systems, Vol. 18/5, Pages: 627-637 (2002).

27. Aliabadi S, Abedi J, Zellars B. Parallel Finite Element Simulation of Mooring Forces on Floating Objects. International Journal for Numerical Methods in Fluids; 41:809-822 (2003).

28. Gresho PM, Sani RL. Incompressible flow and the finite element method, volume two: isothermal laminar flow, John Wiley and Sons, 2000.

29. García JJ, Valls A. ODD Level Set: A New Method for Simulation of Free-surface Problems. Journal of Maritime Research, Vol. III, No. 1, pp 27-40, 2006.

30. Tu S, Aliabadi S, Johnson AA, Watts MD. A robust parallel implicit finite volume solver for high-speed compressible flows. AIAA paper 2005–1396, the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2005.

31. Eisenstat SC. Efficient implementation of a class of preconditioned conjugate gradient methods, SIAM Journal on Scientific and Statistical Computing. 1981; 2:1–4.

32. Tu S, Aliabadi S. A Slope Limiting Procedure in Discontinuous Galerkin Finite Element Method for Gas Dynamics Applications, International Journal of Numerical Analysis and Modeling, V2: 163-178, 2005.

33. Martin JC, Moyce WJ. An experimental study of the collapse of a liquid column on a rigid horizontal plane, Phil. Trans. Roy. Soc. Lond. A244 (1952) 312-324.

34. Iafrati A, Mascio AD, Campana EF. A level set technique applied to unsteady free surface flows, Int. J. Numer. Mech. Fluids 2001; 35; 281-297.

35. Duncan JH. The breaking and non-breaking wave resistance of a two-dimensional hydrofoil, J. Fluid Mech.126, 507-520, 1983.

36. Saad Y., Iterative Methods for Sparse Linear Systems. 2nd edition, 2000.

37. Sharov D., Luo H., Baum J. D. and Lohner R., Implementation of Unstructured Grid GMRES+LU-SGS Method on Shared-Memory, Cache-Based Parallel Computers, AIAA Paper 2000-0927.

38. Karypis G. and Kumar V., Metis 4.0: Unstructured graph partitioning and sparse matrix ordering systems, Technical report, 1998.

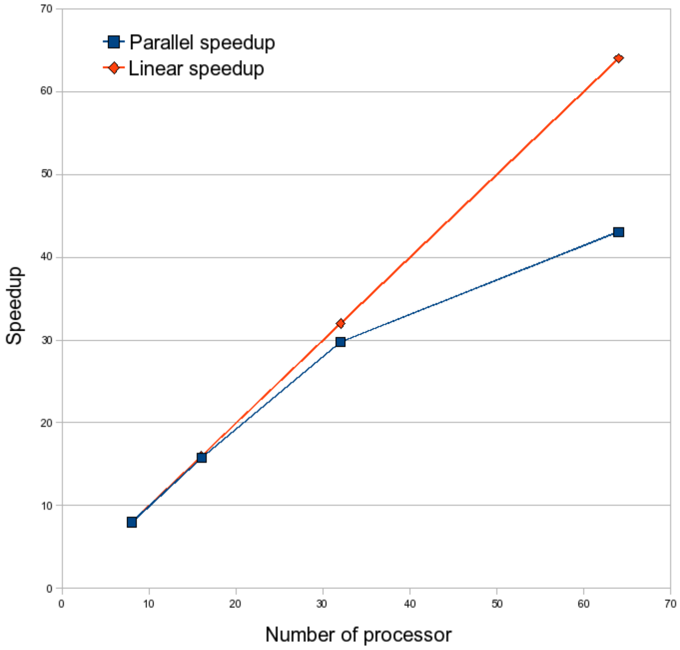


Figure 1. Scalability test result of the CaMEL^H solver.

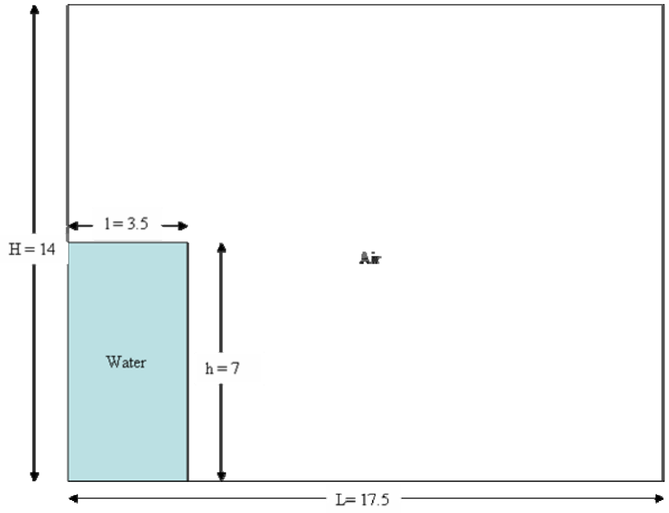


Figure 2. Breaking Dam. Computational domain and the problem definition.

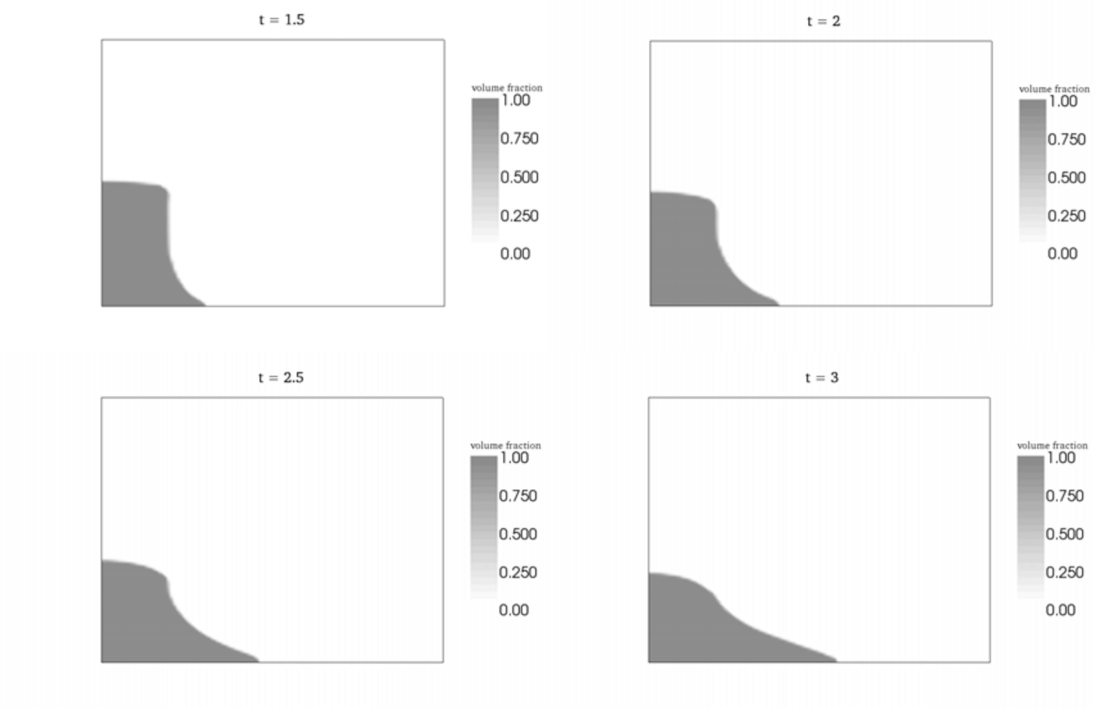


Figure 3. Breaking Dam. The images show the volume of the fluid and the interface between air and water at four instances.

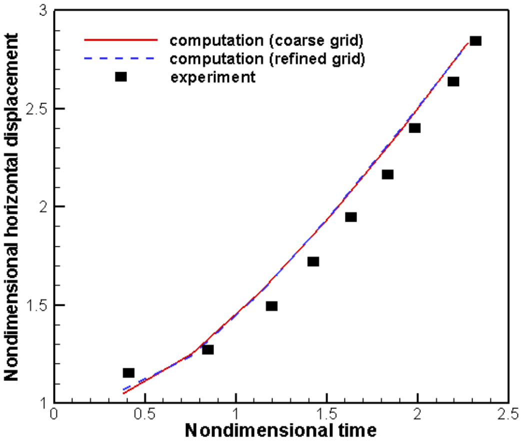


Figure 4. Breaking Dam: Horizontal displacement of the interface front.

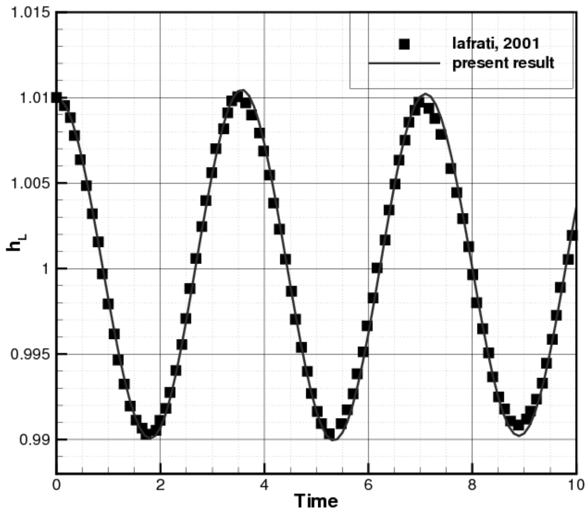


Figure 5. Oscillating Flow in a Tank. Time history of the air-water interface elevation on the left wall of the tank.

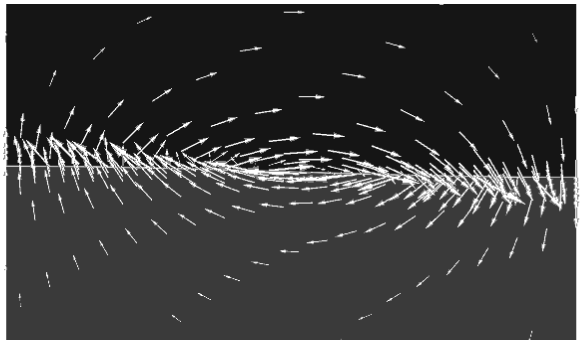


Figure 6. Oscillating Flow in a Tank. Vector plot of the velocity around the interface at $t = 6.75$.

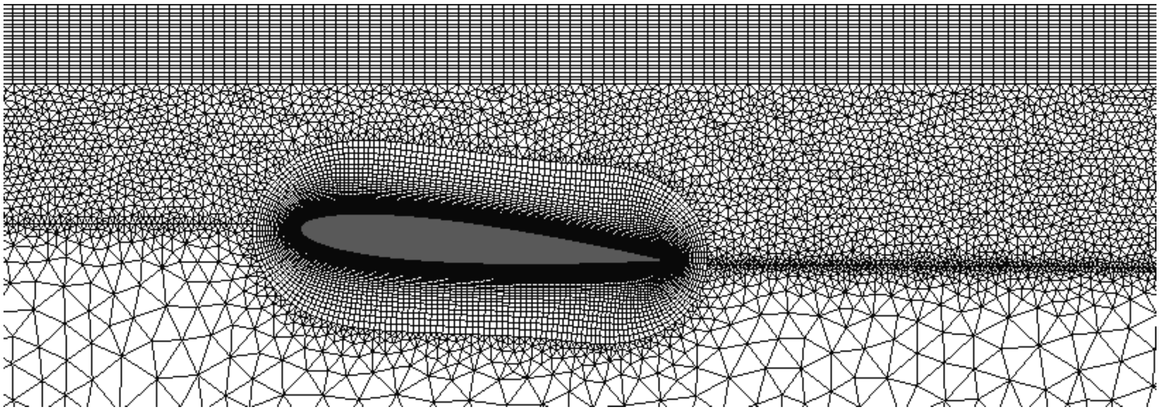


Figure 7. NACA 0012 Submerged Hydrofoil. The hybrid mesh near the hydrofoil.

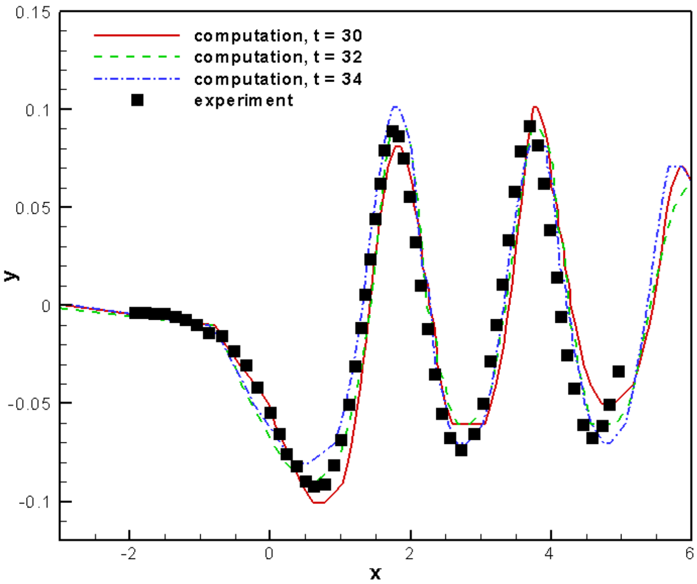


Figure 8. NACA 0012 Submerged Hydrofoil. The interface wave profile at three instances compared with experimentally data.

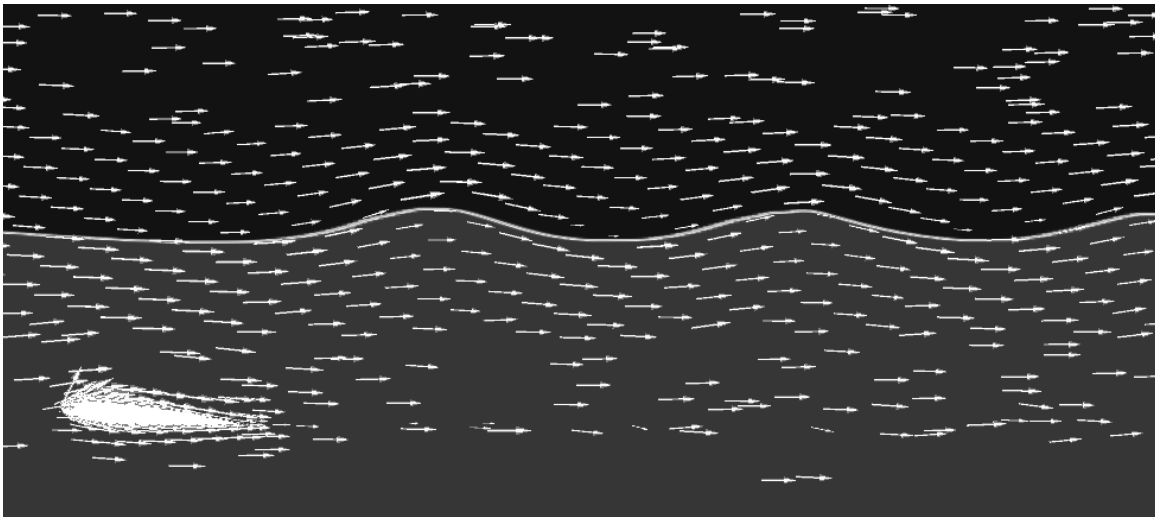


Figure 9. NACA 0012 Submerged Hydrofoil. Velocity vectors near the interface between the air and water.

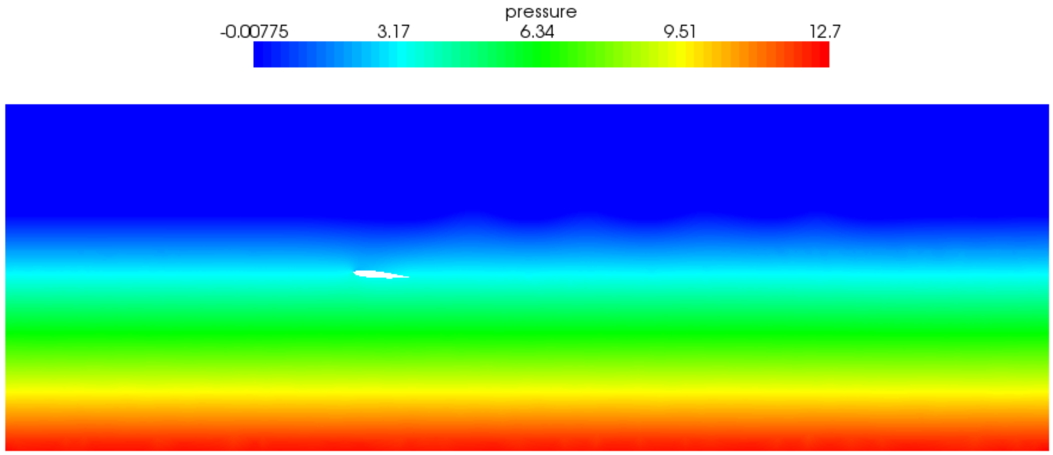


Figure 10. NACA 0012 Submerged Hydrofoil. The image shows the pressure distribution.

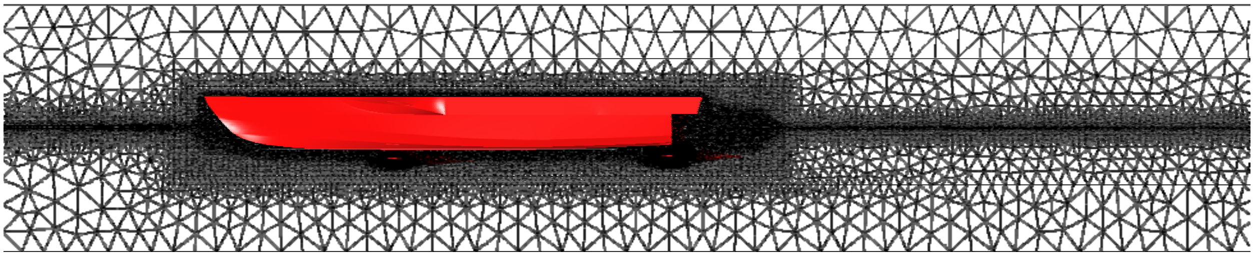
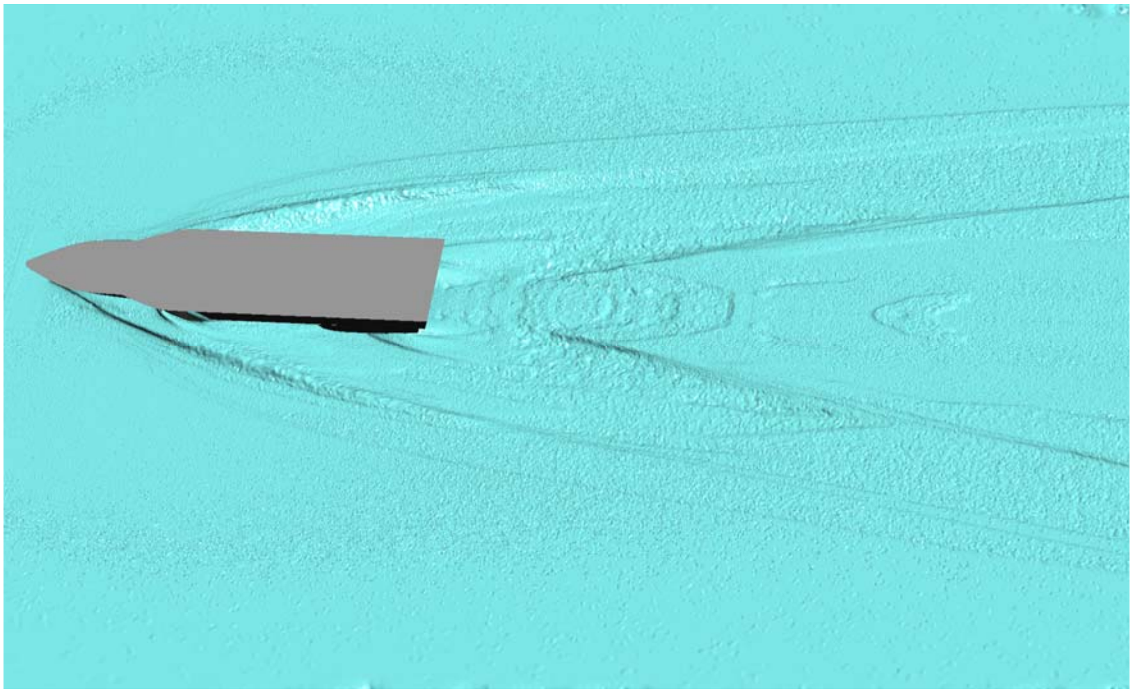


Figure 11. Computational mesh on the symmetry plane. Note that the mesh is clustered near the vehicle and the free surface. The size of the mesh is around 25.2M elements.



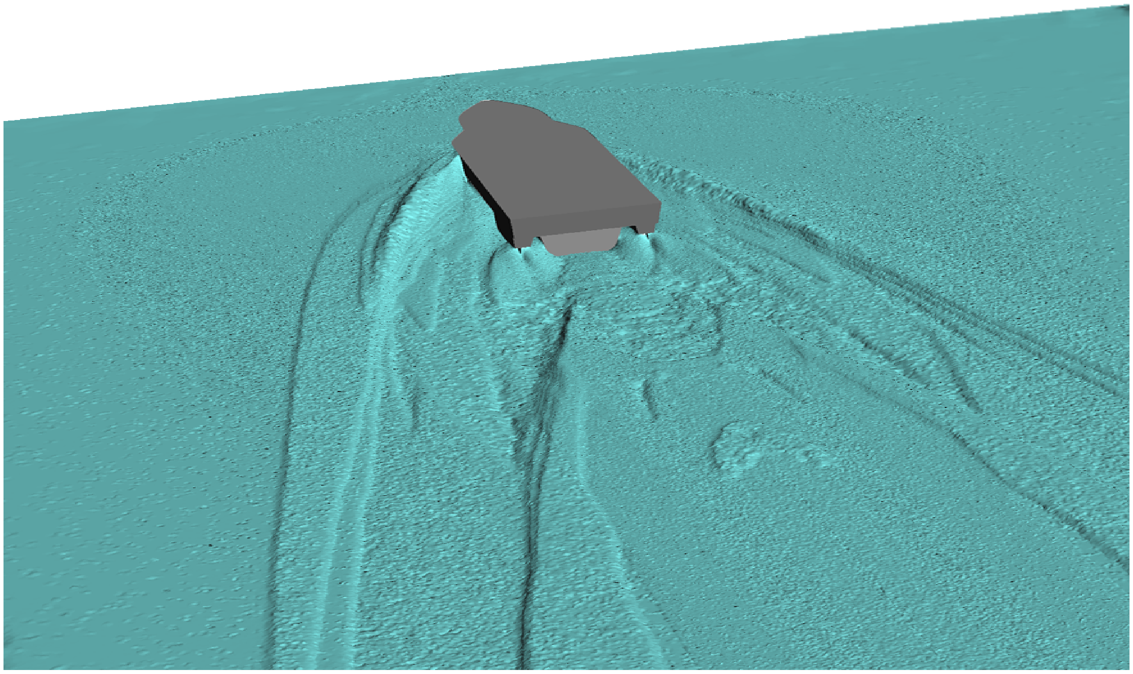


Figure 12. Isosurface of water for the CHSV example case. Top: side view; Bottom: rear view.

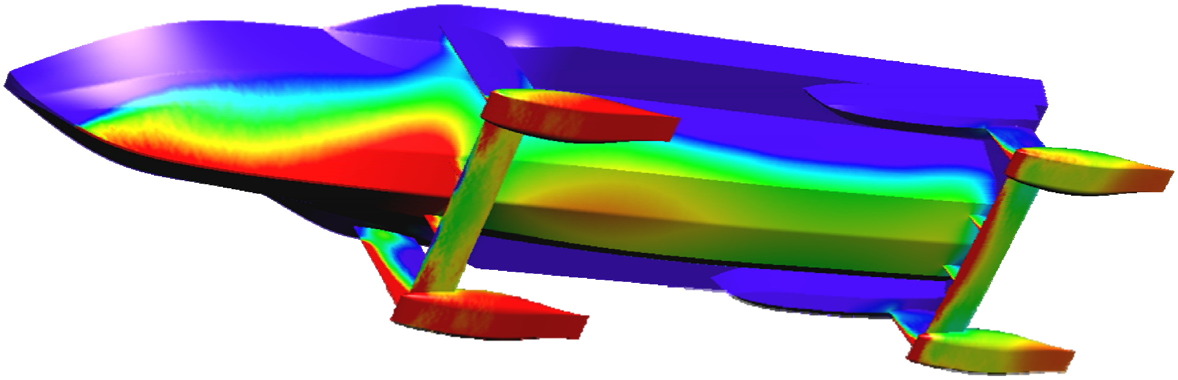


Figure 13. Pressure contours on the vehicle body.

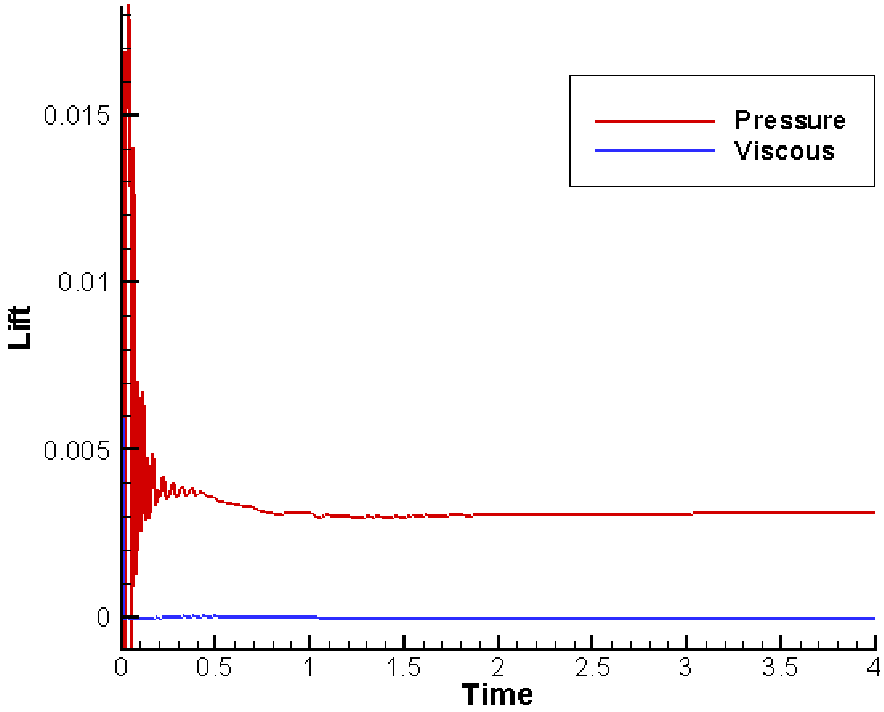


Figure 14. Time history of dimensionless lift force. The red line is the lift force contributed by the pressure, and the blue line is the viscous part of the lift force.