Research Article

Unsteady 2D and 3D Navier-Stokes Solver with Application of Multigrid Scheme to Pressure Poisson Fractional Step on Arbitrary Unstructured Grids in Various Applications with Emphasis on Ship Motion

Mehdi Pourmostafa and Parviz Ghadimi

Department of Marine Technology, Amirkabir University of Technology, Tehran, Iran

Correspondence should be addressed to Parviz Ghadimi; pghadimi@aut.ac.ir

Received 7 July 2020; Revised 14 August 2020; Accepted 28 August 2020; Published 25 October 2020

Academic Editor: Carlo Bianca

Copyright © 2020 Mehdi Pourmostafa and Parviz Ghadimi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A 3D unsteady computer solver is presented to compute incompressible Navier-Stokes equations combined with the volume of fraction (VOF) method on an arbitrary unstructured domain. This is done to simulate fluid flows in various applications, especially around a marine vessel. The Navier-Stokes solver is based on the fractional steps method coupled with a finite volume scheme and collocated grids by which velocity components and pressure fields are defined at the center of the control volume. However, the fluxes are defined at the midpoint on their corresponding cell faces. On the other hand, the CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) scheme is applied to capture the free surface. In the presented fractional step method, the pressure Poisson equation suffers from poor convergence rate by simple iterative methods like Successive Over-relaxation (SOR), especially in simulating complex geometries like a ship with appendages. Therefore, to accelerate the convergence rate, an agglomeration multigrid method is applied on arbitrary moving mesh for solving pressure Poisson equation with two well-known cycles, V and W. In order to maintain accuracy, the geometry details should not change in grid coarsening procedure. Therefore, the boundary faces are assumed to be fixed in all grids level. This assumption requires nonstandard cells in coarsening procedures. To investigate the performance of the applied algorithm, various flows including one and two-phase flows are studied in two and three dimensions. It is found that the multigrid method can speed up the convergence rate of fractional step twofold. In most cases (not all), W cycle displays better performance. It is also concluded that the efficiency of the cycle depends on the number of meshes and complexity of the problem and this is mainly due to the data transferring between grids. Therefore, the type of cycle should be selected judiciously and carefully, while considering the mesh size and flow properties.

1. Introduction

Iterative methods have many applications in engineering problems especially in Computational Fluid Dynamics (CFD). It is found that, in CFD problems, the accuracy is directly related to the number of grids in computational domains. Unfortunately, increasing the mesh number is not always possible, because it directly increases the computational time in common iterative methods. In recent decays, many extensive researches have been conducted to accelerate

the speed of common iterative methods. One of them is the multigrid method which is widely used in CFD codes.

Common single-grid iterative solvers, like SOR and its family, have poor convergence characteristics, especially in finer grids. They are told to produce a smooth error. In other words, they remove high frequencies error easily, but low frequency reduces very slowly. The main idea of the multigrid method is to change the solution to a coarse grid, on which smooth error is rough and low frequency acts like higher ones.

Many researchers used the multigrid method to solve linear systems of equations in order to speed up the solvers. In 1995, Zang and Street [1] applied a 3D finite volume multigrid solver, based on a second-order fractional step scheme, to simulate Navier-Stokes equations. For data transferring, they used biquadratic interpolation between subdomains. In 1996, Meza and Tuminaro [2] used the multigrid method as a preconditioner combined with a conjugated method to solve drift-diffusion equations. Lin [3] in 1994 developed a multigrid method for computing the unsteady inviscid flows on unstructured grids. On the other hand, Liu et al. [4] used a dual time stepping multigrid scheme on structured grids to simulate incompressible flows. Ginzburg and Wittum [5] in 2001 used the V cycle multigrid method as a preconditioner for the biconjugate stabilized method to solve Navier-Stokes equations combined with the VOF method. They used SIMPLE and Gauss-Seidel transforming operators. Subsequently, Tang et al. [6] in 2003 used the same way and applied a V cycle multigrid method to their 3D unsteady incompressible overset-grid solver. Through this method, the data transferring between subdomains takes place on the finest grids. In 2003, Thai and Zhao [7] presented a parallel Navier-Stokes solver based on an unstructured finite volume method, using a multigrid domain decomposition approach. A year later, they [8] successfully applied a high-order finite volume scheme to simulate unsteady incompressible flows, by introducing an unstructured multigrid method. Lambropoulos et al. [9] presented a parallelization of the agglomeration multigrid technique to solve Favre-average Navier-Stokes equations on unstructured grids. The discretization method was finite volume method and FAS (full approximation scheme) was implemented on 2D/3D cases such as airfoil and wing to demonstrate the efficiency of the model. In 2014, Li et al. [10] tried to show that the multigrid method is more effective when the residual restriction satisfies the flux conservation principle. They conducted a study on four convective-diffusion cases and found that the multigrid acceleration is most desirable when the residual restriction satisfies the flux conservation principle. Later on, Lv et al. [11] in 2006 implemented this approach to simulate 3D unsteady compressible flows with arbitrary moving objects. They applied a unique combination of a parallel multigrid method with low Mach number preconditioning. In 2007, Ben Cheikh et al. [12] investigated the efficiency and accuracy of the multigrid method to solve Navier-Stokes/Boussinesq equations. They applied the AFMG (Accelerated Full Multigrid Method) for the solution of an 8:1 differentially heated cavity. They showed that AFMG could speed up the CPU time up to 251 times, related to RBSOR (Red and Black Overrelaxation). Griffith [13] presented a 2D unsplit, staggered discretization of incompressible Navier-Stokes equations with an efficient solver for a system of linear equations. He used the Krylov subspace method with the projection method preconditioner. The preconditioner employed CG (conjugated gradient) method with a geometric multigrid method. He validated his model with several problems such as the lid-driven cavity at various Reynolds numbers. In 2010, Santhosh et al. [14] presented a three-dimensional, transient, and incompressible Navier-Stokes solver. The discretization method is a fractional step method

which is second-order time accuracy and the multigrid method assists its Poisson solver. To show the accuracy and efficiency of the code, 3D lid-driven cavity was simulated by the 4-level V cycle multigrid method. Zong-zhe et al. [15] in 2011 developed a new agglomeration multigrid method to generate coarse grids, automatically. They implement their algorithm to solve compressible Navier-Stokes equations and test that on NACA0012 at Reynolds number $Re = 5000$. In 2013, San and Staples [16] presented a coarse grid projection method to accelerate incompressible flow. This method was introduced to apply to problems including the Poisson equation. They applied the V cycle of their computational method to several 2D/3D test cases in order to show its efficiency. They found that computational time decreases by increasing the distortion ratio on non-Cartesian grids. Langer [17] in 2014 developed a finite volume model with an agglomeration multigrid method to calculate the steady solution of compressible RANS equations on unstructured grids. He used the Gauss-Seidel method to solve the linear systems of equations. Implementing the presented model on two/three-dimensional examples, he showed a wide range of applicability of the solution. In 2014, Cheng and Samtaney [18] developed a high-resolution code to simulate large eddy simulation of incompressible turbulent boundary layers over a flat plate. They used a fourth-order multigrid method to solve the modified Helmholtz equation and the Poisson equation. In 2016, Stiller [19] developed a multigrid method for two-dimensional grids in order to apply the Galerkin formulation of the Poisson equation. He proposed his model to be used in more complicated applications such as CFD. Lygidakis et al. [20] in 2016 presented an agglomeration multigrid methodology which was applied on two academic CFD codes: GALATEA and GALATEA-I. These two codes are able to simulate compressible and incompressible flows on unstructured grids. Comparing different approaches of multigrid methodology, they found that, in general, higher acceleration would archive in incompressible flow compared with the compressible flow. That is due to the elliptical nature of incompressible equations. In 2017, Matthioudakis et al. [21] developed a fourth-order finite-difference model on cell-centered grids to calculate Navier-Stokes equations. They introduced a new high-order transfer operator to apply the multigrid method on pressure correction. In their work, a comparison was made between different multigrid cycles. They showed that the accuracy of the method is retained in time and space by increasing the Reynolds number. Abide and Zeghamati [22] in 2019 investigated a multigrid defect correction to solve a fourth-order compact scheme of a singular Poisson equation. They applied the presented method to the Hodge-Helmholtz decomposition test case and showed better accuracy related to second-order discretization. In 2019, Liu et al. [23] proposed a mode multigrid method and applied it to a steady-state solver for unstructured grids. They used the solver for 2D and 3D foils and showed that the method is 3 to 6 times faster than the baseline method while ensuring computational accuracy. Ha et al. [24] in 2019 presented an interpolation operator on a multigrid method based on distance weight. The performance of the method was compared with that of an interpolation operator by area (volume) intersection

for the finite volume discretization. By solving heat conduction problems on 2D and 3D domains, they concluded that the distance weight operator is faster and more reliable than the area intersection approach.

1.1. Background and This Paper. A three-dimensional Navier-Stokes solver is developed based on a fractional step method which was introduced by Kim and Choi [25] and Panahi et al. [26]. The finite volume method was used to discretize the equations on unstructured grids. The solver is combined with CICSAM [27] volume of fraction method to capture the free surface elevation in various applications, especially in ship motions. By applying the body-attached mesh and calculating the pressure field in the computational domain, hydrodynamic forces and moments can be calculated on the ship hull [28]. As pointed out, the common single-grid iterative solvers have poor convergence characteristics. In this paper, the agglomeration multigrid method is applied to the Poisson solver of fractional steps in order to speed up the iterative solver. For this purpose, a fully unstructured database is established for the solver. The finest grid is introduced as an input to the algorithm (it could be generated by an external mesh generated software like Gambit). The coarse grids are automatically generated through the presented agglomeration algorithm. This method was initially introduced by Smith [29] and Lallemand et al. [30]. In this paper, in order to generate coarse grids with better performance, a new coarsening method is applied and for more accuracy, a rather complicated data structure is designed to save all faces of the coarsened cell. These non-standard cells are used to save the geometrical details of the ship like rudder and brackets. In order to verify the accuracy of the solver and investigate its performance, various 2D and 3D test cases are simulated. The acceleration of V and W multigrid cycle is also compared in all simulations.

2. Governing Equations

Two main governing equations for unsteady, incompressible Newtonian fluid motion are momentum and conservation equations. These equations in three-dimensional space are written as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + g_i, \quad (1)$$

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2)$$

where u is the velocity, ρ the density, P the pressure, and g_i the gravitational acceleration. In the cases where there are two-phase fluid simulations, one should solve the volume fraction transport equations as in

$$\frac{\partial \alpha}{\partial t} + \vec{\nabla} \cdot (\alpha \vec{u}) = 0. \quad (3)$$

In this equation, α is a fractional value which demonstrates the amount of water in the cell:

$$\alpha = \begin{cases} 1, & \text{Cell inside water,} \\ 0, & \text{Cell inside air,} \\ 0 < \alpha < 1, & \text{for transitional area.} \end{cases} \quad (4)$$

Generally, equations (1)–(3) should be solved simultaneously. For this purpose, the fluid density and viscosity of each cell in the computational domain can be expressed as a function of the volume fraction α as follows:

$$\begin{aligned} \rho &= \rho_1 \alpha + \rho_2 (1 - \alpha), \\ \mu &= \mu_1 \alpha + \mu_2 (1 - \alpha), \end{aligned} \quad (5)$$

where the subscripts 1 and 2 denote water and air, respectively.

3. Discretization

3.1. Discretization of Navier-Stokes Equations. As previously pointed out, the finite volume method is applied to discretize the governing equations including Navier-Stokes equations. Integrating Navier-Stokes equations over each control volume can lead to

$$\begin{aligned} \frac{d}{dt} \int_V \vec{u} dV + \int_A \vec{u} (\vec{u} \cdot \vec{n}) dA \\ = \int_A \nu \vec{\nabla} \cdot \vec{u} \cdot \vec{n} dA - \frac{1}{\rho} \int_A \vec{\nabla} P dV + \int_V \vec{g} dV. \end{aligned} \quad (6)$$

For the fluid velocity u_i , the diffusion term (the first term on the right-hand side of equation (6)) is discretized using overrelaxed interpolation Jassak [31]:

$$\int_A \nu \vec{\nabla} u_i \cdot \vec{n} dA = \int_V \nu \vec{\nabla} \cdot (\vec{\nabla} u_i) dV = \sum_{\text{faces}} \nu_f \vec{A}_f \cdot (\vec{\nabla} u_i)_f. \quad (7)$$

In order to discretize the convection term (the second term in the left-hand side of equation (6)), fluid velocity at the control volume faces is needed to compute fluxes. This can be evaluated as follows:

$$\int_A u_i (\vec{u}_i \cdot \vec{n}) dA = \sum_{\text{faces}} u_f F_f. \quad (8)$$

To calculate u_f , the gamma interpolation scheme is used which was introduced by Jassak [31].

It must be mentioned that the Crank-Nicholson scheme is used for time discretization of diffusion and convection terms in equation (6). This scheme is of second-order accuracy and increases stability.

The pressure term (second term on the right-hand side of equation (6)) is discretized as

$$\int_A P \vec{n} dA = \sum_{f=CV_f \text{ faces}} P_f \vec{A}_f, \quad (9)$$

where A_f is the direction component of the face area vector.

Using the common linear interpolation between two neighboring control volume centers results in severe

oscillations in the velocity field. This is of great importance, especially when there are two fluids with high-density ratios, for example, water and air in this paper. Here, a Piecewise Linear Interpolation is used which was developed by Panahi et al. [26].

3.2. Discretization of VOF Equation. The finite volume discretization of volume fraction transport equation (3) is based on the integration over the control volumes and the time step:

$$\int_t^{t+\delta t} \left(\int_V \frac{\partial \alpha}{\partial t} dV \right) dt + \int_t^{t+\delta t} \left(\int_V \nabla \cdot \alpha \vec{u} dV \right) dt = 0. \quad (10)$$

The first term is a simple integral form. By applying the Gauss theorem on the second term and assuming a small variation of F_f , it can be shown that

$$(\alpha_p^{t+\delta t} - \alpha_p^t) \frac{V_P}{\Delta t} = \sum_{f=1}^{\text{faces}} \frac{1}{2} (\alpha_f^t + \alpha_f^{t+\delta t}) F_f \delta t, \quad (11)$$

where (α_f) in equation (11) must be approximated on the face of each control volume. Simple interpolation results in nonphysical values. This leads to using a high-order composite interpolation. In this paper, CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) scheme is applied [27]. CICSAM uses CBC (Convection Boundedness Criteria) $(\tilde{\alpha}_{f_{\text{CBC}}})$ and UQ (Ultimate Quickest) $(\tilde{\alpha}_{f_{\text{UQ}}})$ interpolation by introducing a weighting factor γ_f [32, 33]. This factor is evaluated according to the slope of the free surface and the direction of fluid motion [27]. Based on NVD (Normalized Variable Diagram), normal face value is obtained as follows [32]:

$$\tilde{\alpha}_f = \gamma_f \tilde{\alpha}_{f_{\text{CBC}}} + (1 - \gamma_f) \tilde{\alpha}_{f_{\text{UQ}}}. \quad (12)$$

NVD definitions are shown in Figure 1 [26] where donor, acceptor, and upwind cells are defined according to flow direction for each control volume face:

$$\tilde{\alpha}_f = \frac{\alpha_f - \alpha_U}{\alpha_A - \alpha_U}. \quad (13)$$

Substituting equation (13) into equation (12) results in the estimation of (α_f) .

In the VOF approach, CFL (Courant–Friedrichs–Lewy) number plays an important role in the stability of the solution. The Courant number is calculated according to equation (14) for each cell and is set to be less than 0.2 for all two-phase flow test cases [28]:

$$\text{Courant number} = \sum_{\text{All faces}} \max \left(-\frac{\text{Flux} \cdot dt}{\text{Cell volume}}, 0 \right). \quad (14)$$

As mentioned before, a body-attached mesh is used to simulate the body motions. In this way, the relative velocity should be applied in all equations as follows:

$$\vec{u} = \vec{u}_{\text{fluid}} - \vec{u}_{\text{body}}, \quad (15)$$

where $\vec{u}_{\text{fluid}}$ is fluid velocity vector and $\vec{u}_{\text{body}}$ is the velocity of the moving body (which is equal to mesh velocity).

FIGURE 1: Flow direction determines the donor, acceptor, and upwind cells [26].

4. Numerical Algorithm

The main algorithm which is used to couple the pressure and velocity field is the modified fractional steps which were firstly proposed by Kim and Choi [25]. Consider the Navier-Stokes equations for the velocity component u_i in its discretized form on a given control volume. The first step is to solve the first intermediate velocity $(\hat{u}_i)$ with the pressure gradient in the previous time step $(G_i(P^n))$:

$$\frac{\hat{u}_i - u_i^n}{\Delta t} = \frac{1}{2} [H(u_i^n) + H(\hat{u}_i)] - \frac{1}{\rho} G_i(P^n) + K_i, \quad (16)$$

where

$$\begin{aligned} H(u_i) &= \int_A \nu \frac{\partial u_i}{\partial n} dA - \int_A u_i^n \vec{U}_f \cdot \vec{n} dA, \\ G_i(P) &= \int_A P n_i dA, \\ K_i &= \int_V g_i dV. \end{aligned} \quad (17)$$

Then, the second intermediate velocity is calculated as

$$u_i^* = \hat{u}_i + \frac{\Delta t}{\rho} G_i(P^n). \quad (18)$$

Using equation (18) results in the pressure Poisson equation which gives a new pressure field P^{n+1} :

$$\oint_{A\rho} \frac{1}{\rho} \frac{\partial P^{n+1}}{\partial n} dA = \frac{1}{\Delta t} \oint_A u_i^* dA, \quad (19)$$

$$\frac{\partial P^{n+1}}{\partial n} \Big|_{\text{Boundary}} = 0. \quad (20)$$

By writing equation (19) for all cells in the computational domain, a system of equations in the form $(AX = B)$ is obtained. This matrix is solved by applying the multigrid method which will be explained in the “Multigrid Method.” After that, the velocity at the new time step is computed as follows:

$$u_i^{n+1} = u_i^* + \frac{\Delta t}{\rho} G_i(P^{n+1}). \quad (21)$$

Since the new pressure is used to calculate (in other words, correct) velocities, global mass conservation is automatically satisfied. After the calculation of all velocity components in an outer iteration, the control volume face velocity vector is calculated for the next time step from

equation (22). It includes the effect of pressure gradient on the calculation of the face velocity vector to overcome the check-board pressure in the nonstaggered (collocated) arrangement:

$$\vec{U}_f^{n+1} = \text{LI}(u_i^*) + \left(\frac{\Delta t}{\rho} \frac{\partial P^{n+1}}{\partial n} \right)_f \vec{n}. \quad (22)$$

The overall algorithm is shown in Figure 2. Also, more details on the algorithm can be found in a paper by Panahi et al. [28].

5. Calculating Forces, Moments, and Rigid Body Motions

As pointed out earlier, the body-attached mesh (in which mesh and body move together) is used in order to simulate the motion of rigid bodies, like ship hull (Figure 3).

Solving the Navier-Stokes equations, one can calculate the flow forces and moments by integrating the pressure and viscous stress over the body boundaries (like ship hull). These forces ($\vec{F}$) and moments ($\vec{M}$) cause the body to move in 6 degrees of freedom. These movements could be calculated by solving the linear and angular momentum equations (equations (23) and (24)):

$$\int_{\text{Body cells}} \vec{F} = m \vec{a}, \quad (23)$$

$$\int_{\text{Body cells}} \vec{M} = I \vec{\alpha} + \vec{\omega} \times I \vec{\omega}, \quad (24)$$

where (I , α and ω) are body inertia matrix, body angular acceleration, and body angular velocity, respectively.

6. Multigrid Method

As previously pointed out, conventional iterative single-grid methods like SOR for solving Navier-Stokes equations on fine grids suffer from poor convergence characteristics. These solvers remove high frequencies and short wavelengths, so they are known to have smooth property. Multigrid methods effectively remove low-frequency error components on finer grids, by converting the results to coarse grids. In this paper, the finest grid for the computational domain can be generated by any commercial software such as Gambit or ANSYS. Then, coarse grids would be generated based on the coarsening algorithm.

6.1. Coarsening Algorithm. The coarsening algorithm is based on the method which Okamoto et al. [34] suggested in 1998. The main steps of the algorithm are described as follows:

- (1) The starting vertex in the fine grid should be selected (e.g., vertex no. 1 in Figure 4)
- (2) In the second step, all the cells which contain the sharing vertex are merged
- (3) The next vertex is selected (e.g., vertex no. 2 in Figure 4)

FIGURE 2: Fractional step method algorithm.

FIGURE 3: Schematics of the body-attached mesh of Research Vessel (RV).

- (4) If one of the sharing cells (sharing cells of a vertex: the cells which the vertex is one of its nodes) of this vertex (selected in step 3) is used in previous coarsening, then go to step 3 and select another vertex; else go to step 2

Figure 4 shows the above algorithm for a simple 8-cells grid. In this example, after merging all the sharing cells of vertex number 1, vertex number 2 is passed and vertex number 3 is selected for the next coarsening. Figure 4 shows a very simple cell arrangement while in reality, it is not always the case.

A slightly more complicated cell arrangement is displayed in Figure 5. In this case, after coarsening, two larger cells are produced. Figure 5(a) shows the larger cells with simplified faces. Although standard unstructured cells (Hexahedron, Wedge, Tetrahedron, and Pyramid are referred to as standard meshes) are produced in this method, eliminating the details may compromise the accuracy, especially near the important boundaries like no-slip wall (ship hull). In Figure 5(b), the coarsening is done by creating

FIGURE 4: The applied coarsening algorithm in simple cell arrangement.

FIGURE 5: Coarsening algorithm with standard and polyhedron cells.

polyhedron cells and without eliminating the details. In this method, a more complicated data structure is needed in order to deal with arbitrary polyhedron cells with an arbitrary number of faces. (In coarse grids, there is more than one face between two specific cells.) Although it is already known that this type of data structured requires more memory allocations which is discussed in the following sections. This method is more applicable in complex three-dimensional geometries like simulating ships with all appendages.

For better coarsening and efficient simulation, it is recommended that the vertexes not be chosen, randomly. Therefore, before the coarsening, a list of vertexes that have more sharing cells must be made, and the coarsening should follow according to the list. With this recommendation, the highest efficiency in grid coarsening is achieved. Based on the explained algorithm, three or four levels of coarse grids can be generated due to the size of the computational domain.

6.2. Multigrid Algorithm. As mentioned before, the multigrid technique is used to damp the low-frequency error components. Based on the number of coarse grids, the multigrid method can be applied. Here, for simplicity, a two-grid scheme is introduced. The implementation of the algorithm for more grids is very similar.

Assume the linear system of equation ($Ax = B$). The two-grid (multigrid) scheme to solve this system of equation is as follows [35]:

- (1) Relax with ($A_1x_1 = B_1$) on fine grid (with common iterative methods like SOR) and find the

approximate solution v_1 (relax until the convergence rate decreases)

- (2) Compute the fine grid residual ($r_1 = f_1 - A_1v_1$) and restrict it to the coarse grid
- (3) Solve ($A_2e_2 = r_2$) on the coarse grid
- (4) Prolong the coarse grid error to the fine grid and correct the fine grid approximation ($v_1 = v_1 + e_2$)
- (5) Relax with ($A_1x_1 = B_1$) on fine grid with initial guess v_1

The subscripts 1 and 2 denote the fine and coarse grid, respectively. This algorithm can be extended on more grids and in different cycles. Figure 6 illustrates the structure of one iteration step (cycle) of a multigrid method with a few pictures. The cases $\gamma = 1$ and $\gamma = 2$ are usually used. For obvious reason, the cases $\gamma = 1$ and $\gamma = 2$ refer to V and W cycles, respectively (Figure 4). The number γ is also called a cycle index [36]. In this paper, V and W cycles are used for comparison in all simulations.

6.3. Restriction and Prolongation Operators. As mentioned in the previous section, the multigrid method requires some operators to transfer data between the grids. The prolongation operator transfers a set of variables from a coarse grid to a fine grid and the restriction operators transfer a set of variables from a fine grid to a coarse grid. Many different operators are defined for prolonging and restriction. In this paper, the operators which are introduced by Ferziger and Peric [37] are used:

FIGURE 6: Structure of one multigrid cycle for different numbers of grids and different values of the cycle index γ [36].

Restriction operator:

$$\phi_C = \frac{1}{N_f} \sum_{i=1}^{N_f} \left[\phi_{F_i} + \overrightarrow{(\text{grad } \phi)_{F_i}} \cdot \overrightarrow{(r_C - r_{F_i})} \right]. \quad (25)$$

Prolongation operator:

$$\phi_{F_i} = \phi_C + \overrightarrow{(\text{grad } \phi)_C} \cdot \overrightarrow{(r_{F_i} - r_C)}. \quad (26)$$

In equations (25) and (26), r is the cell center of the control volume, (N_f) is the number of the fine grid control volume, and f and c denote the fine and coarse grid, respectively. Figure 7 clearly shows the operators.

7. Numerical Results

Based on the algorithm described above, a 3D numerical code is developed and its accuracy is verified and different simulations are performed. In each case, different multigrid cycles are applied and the computational times are compared to see which cycle provides better efficiency in each case. For better comparison, two different parameters are considered: residual and computational time. The residual of each iteration for the Poisson equation is defined as follows:

$$e_2 = \frac{\left\{ \sum_{g=1}^n e_g^2 \right\}^2}{n}. \quad (27)$$

7.1. 2D Triangular Lid-Driven Cavity. Firstly, a two-dimensional test case is investigated. The lid-driven cavity is a very common test case to check the accuracy of Navier-

FIGURE 7: Transfer of variables from fine to coarse grid and vice versa [37].

Stokes equations (one-phase flow). In this case, the domain is a triangular shape, as shown in Figure 8. The upper side is moving with unit velocity while the other two sides are assumed as no-slip walls. Meanwhile, 28,731 triangular unstructured cells are used to discretize the computational domain. A section of the meshed domain (section z-z in Figure 8) is shown in Figure 9. The Reynolds numbers are set to be $(Re = (Vl/\nu))$. This is a steady-state test case, so the simulation should be performed until the velocity and pressure field become steady.

Velocity fields are plotted along two horizontal and oblique lines. The horizontal line joins the center of AB to the center of AC and the oblique line joins point A to the center of BC. Results are compared with the computational results of Jagannathan et al. [38] in Figures 10 and 11.

FIGURE 8: Schematics of the domain of a triangular lid-driven cavity.

FIGURE 9: The z-z section with triangular mesh.

FIGURE 10: Velocity along the line joining A point to the center of BC.

Two V and W cycles are used for simulating in 4-grid size. The number of cells in four different grid sizes is shown in Table 1. The time step is set to 0.05 sec and therefore in each time step, the multigrid method is used to solve the pressure Poisson equation as described in the previous

sections. Iteration continues until $(\text{Log}(\text{Residual}) < -8.5)$ is achieved.

The residual reduction graphs in each time step are almost the same. Yet, this graph is plotted at the second time step in Figure 12. In this figure, VMG3 implies V cycle

FIGURE 11: Velocity along the line joining center of AB to the center of AC.

TABLE 1: Different grid size levels (as the cells are not standard unstructured cells, so it is not possible to show them graphically) in the triangular cavity.

Grid size level	Number of cells
Finest grid	28,731
Coarse grid level 1	6260
Coarse grid level 2	1572
Coarse grid level 3	450

TABLE 2: Convergence features for a triangular lid-driven cavity for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	3090	1	215,000
3-grid V cycle	1955	1.58	279,000
3-grid W cycle	1075	1.82	279,000
4-grid V cycle	659	1.63	281,000
4-grid W cycle	348	1.89	281,000

FIGURE 12: Residual reduction for the second time step in the triangular cavity.

multigrid method using a 3-grid size. As observed in Figure 12, the SOR scheme slightly reduces the residual while in the multigrid method, the residual reduces much faster. For example, 4400 iterations in WMG4 are almost equal to 25,000 iterations in SOR.

The CPU time and memory allocations of different iterative methods (for 50 time steps) are compared in Table 2. In this case, VMG4 and WMG4 do not offer impressive performance in reducing residuals; it seems that their difference with VMG3 and WMG3 is very small.

Based on these two parameters, the W cycle with 4 grids offers better efficiency. This cycle decreases the simulation time twofold.

7.2. Backward-Facing Step. In this case, 2D flow over a step in a pipeline is investigated known as BFS (Backward-Facing Step) problem. The schematic geometry of the problem is shown in Figure 13. At the inlet boundary condition, a fully developed flow is applied. The Reynolds number, in this case, is computed ($Re = (VD/\nu) = 200$). D is the pipe diameter. A fully structured grid is used as the finest grid level.

Due to the Reynolds number, two back warding flows are developed at the top and bottom of the pipe, as shown in Figure 13. There are 30,400 structured cells used for simulations. When the fluid flow becomes steady, the velocity profile at two sections ($x = 6, 14$) is plotted and compared by similar works [39, 40] (Figures 14 and 15).

For better comparison, it is also necessary to calculate the length of the backward flow in the pipeline (X_1, X_2, X_3). These lengths are compared with those in Ertruk's work [40] displayed in Table 3. Results indicate good agreement in simulation.

In this test case, a three-grid level V and W cycle is used for the simulations. Table 4 shows two coarsened grid size level. Since coarse grid level 2 consists of 771 cells, more coarsening is not effective in the solution. Therefore, 3 grid levels are used in multigrid cycles.

Using these grid sizes, the residual reduction is shown in Figure 16.

The computational times are also compared in Table 5 for 50 time steps.

Usually, the WMG3 cycle reduces the error in fewer iterations, but in this particular case, the VMG3 operates with a higher speed. This is because the number of cells is very low and with prolonging once, error oscillation in

FIGURE 13: Schematics of the geometry for the BFS.

FIGURE 14: Horizontal velocity at $(x = 6)$. FIGURE 15: Horizontal velocity at $(x = 14)$ (Gartling [39]).TABLE 3: Comparison of x_i with similar work.

	X_1	X_2	X_3
Ertruk [40]	5.917	4.738	10.2
Present study	5.8	4.7	10.1

TABLE 4: Different grid size level in BFS.

Grid size level	Number of cells
Finest grid	11,400
Coarse grid level 1	3055
Coarse grid level 2	771

FIGURE 16: Residual reduction for the second time step in BFS.

TABLE 5: Convergence features for BFS for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	1505	1	~47,000
3-grid V cycle	737	2.04	~69,000
3-grid W cycle	764	1.97	~69,000

coarser cells gets sufficiently smooth (V cycle). More usage of coarser cells and transferring more information between the cells (in this particular case) increases the CPU time (W cycle). Therefore, in this case, VMG3 is a better iterative cycle to be selected.

7.3. 3D Cubical Lid-Driven Cavity. Consider the three-dimensional cubical lid-driven cavity with side's length of $(l = 1 \text{ m})$, as depicted in Figure 17. The top wall is moving to the right with constant velocity $(v = 1 \text{ (m/s)})$, while for other faces, the stationary no-slip wall condition is applied. The cavity is filled with a Newtonian fluid (constant viscosity and density). In this case, the simulations are performed at three Reynolds numbers $(\text{Re} = (vl/\nu) = 100, 400, \text{ and } 1000)$.

The computational grid of $50 \times 50 \times 50$ control volumes is considered as a fine grid with $(\Delta t = 0.01 \text{ s})$. The iteration loop is continued until $(\text{Log}(\text{Residual}) < -9)$ is achieved. After the solution becomes steady, the velocity component in X direction (u) is plotted along y -axes at the midplane

FIGURE 17: The geometric model of 3D cubical lid-driven cavity.

($x = z = 0.5$) and compared with a similar work of Jiang et al. [41] and Ku et al. [42] at different Re numbers (Figure 18). The results are in agreement and this shows that a multigrid scheme can predict velocity distribution with good accuracy.

Figures 19–21 compare the pressure contours at different sections ($x = 0.5$, $y = 0.5$, and $z = 0.5$) with Jiang et al. [41] at $Re = 100$, 400, and 1000, respectively.

As pointed out earlier, for solving the pressure Poisson equation, the multigrid method is used. The number of cells in four different grid sizes is shown in Table 6. As observed, the coarsening method can efficiently merge the cells and produce grid level 1 with less than $(1/4)$ cells of the finest grid.

The residual reduction versus iteration for the second time step is illustrated in Figure 22.

Table 7 also shows the CPU time and memory allocations of different iterative methods for 50 time steps.

Based on the numerical results of Table 8, W cycle with 4 grids is used for the entire simulation of a 3D cubical lid-driven cavity. It is clear that, for this case, the multigrid approach can easily increase the speed of computations by 1.85 times.

In order to better evaluate the current solver, a more complicated 3D cavity, two-sided nonfacing lid-driven (3D-TSNFL) is investigated (Figure 23). In this test case, the top wall is moving to the right, while the left vertical wall is moving down with the same constant velocity. All the other parameters are exactly the same as the 3D lid-driven cavity which was mentioned earlier.

The computational domain consists of $39 \times 39 \times 39$ control volumes as a fine grid with ($dt = 0.01$ s). After the solution becomes steady, the velocity components (u and w) are plotted along z - and x -axes at the midplane ($y = 0.5$) and compared with similar work of Ben Beya and Lili [43] (Figure 24).

The velocity contours are also illustrated in Figures 25 and 26 and compared with that provided by Ben Beya and Lili [43].

The number of cells in four different grid sizes is shown in Table 9 which were used in the multigrid method. As

observed, the coarsening method can efficiently merge the cells and produce grid level 1 with less than $(1/4)$ cells of the finest grid.

The residual reduction versus iteration for the second time step is shown in Figure 27.

Table 8 also shows the CPU time and memory allocations of different iterative methods for 50 time steps.

Based on the numerical result in Table 8, W cycle with 4 grids is used for the entire simulation of 3D-TSNFL. It is clear that, for this case, the multigrid approach can easily increase the speed of computations by 1.8 times.

7.4. Collapse of a Water Column with Obstacle. This test case is a three-dimensional two-phase flow problem. The geometry and initial conditions are illustrated in Figure 28. In the first time step, the column of water breaks and flows into the domain. This problem is a very complicated flow and because of the lower density of air (related to water), it gets trapped in the water. Trapped air is subjected to a large buoyancy force and tends to rise up.

The pressure which water exerts on the obstacle along with water height at two different stations of the reservoir is the main parameter which is investigated in this test case.

Based on the computational domain, a fully structured grid is used as the finest grid level, similar to BFS. According to previous cases, a multigrid method is used with a three-grid level for simulation. The grids level size is shown in Table 10.

As observed, the proposed coarsening algorithm has a better performance in fully unstructured grids. Comparison of the coarsening in an unstructured triangular cavity test case with the structured BFS and 3D dam break grids reveals that, in unstructured grids, the coarsening ratio is about 4.5, while in the structured grids, the ratio is almost 3.9 to 4.0. This is due to the fact that, in structured grids, the number of sharing cells for each vertex is less than the unstructured grids. For example, in a triangular cavity, there are vertexes with 6 sharing cells, while this number is 4 in BFS or dam break test case.

The best multigrid cycle is chosen due to the results of the residual reduction in the second time step Figure 29 and

FIGURE 18: Profile of u and v velocity along y - and x -axes for a 3D cubical lid-driven cavity at $Re = 100, 400$, and 1000 .

also the simulation time of the problem for 100 time steps which is shown in Table 11.

In this simulation, it is concluded that W cycle is slightly better than V cycle because of a relatively large number of cells. Hence, W cycle is selected for the entire simulation. The simulation is performed for 6.0 seconds. As shown in Figure 28, the water height is computed at points H1 and H2 and pressure is numerically measured on the obstacle at points P1 to P3. This test case is experimentally conducted at the Marine Research Institute of the Netherland (MARIN) and results were published by Kleefsman et al. in 2005 [44].

Figure 30 shows the time history of water height at H1 and H2 and is compared with experimental measurements.

In both pictures of Figure 30, the agreements are good until the water has returned from the back of the reservoir (after about 1.5 s). After that, although some differences are observed, global behavior is almost still the same. The pressure histogram is also plotted at points P1 to P3 in Figure 31. It is demonstrated that the presented solver can estimate the pressure with acceptable accuracy related to experimental data, especially at points P1 and P2.

FIGURE 19: Pressure contour at different sections for $Re = 100$. (a) Section $x = 0.5$ (left: Jiang et al.; right: present study). (b) Section $y = 0.5$ (left: Jiang et al.; right: present study). (c) Section $z = 0.5$ (left: Jiang et al.; right: present study).

Ultimately, the free surface profile is compared to experimental data in Figure 32. It is shown that CICSAM scheme can capture the free surface shape almost the same as the actual case which makes it a reliable approach in VOF method.

7.5. Barge Resistance. In this section, the resistance of a moving barge is calculated as a simple marine test case. Figure 33 and Table 12 show the main parameters of the barge geometry. Barge resistance was experimentally measured in the towing tank of the marine laboratory at Sharif

FIGURE 20: Pressure contour at different sections for $Re = 400$. (a) Section $x = 0.5$ (left: Jiang et al.; right: present study). (b) Section $y = 0.5$ (left: Jiang et al.; right: present study). (c) Section $z = 0.5$ (left: Jiang et al.; right: present study).

University of Technology [26] at a constant velocity ($V = 0.807 \text{ (m/s)}$).

Due to water forces, the draft and trim of the barge may change. Therefore, the simulation is performed in two degrees of freedom motion which basically means that the

barge is allowed to have pitch and heave motions. A total of 158,891 structured grids are used as the finest grid size to simulate barge motion (Figure 34). No-slip boundary condition is used for barge and outflow boundary condition is applied for other 6 boundaries.

FIGURE 21: Pressure contour at different sections for $Re = 1000$. (a) Section $x = 0.5$ (left: Jiang et al.; right: present study). (b) Section $y = 0.5$ (left: Jiang et al.; right: present study). (c) Section $z = 0.5$ (left: Jiang et al.; right: present study).

Using the multigrid method, the residual reduction for the second time step is illustrated in Figure 35.

The simulation is conducted with all types of cycles for 50 time steps and as a result, the W cycle with a four-grid

level is adopted because of the better CPU speed-up ratio Table 14.

Figure 36 shows the resistance of the barge versus time. After about five sinusoidal oscillations, the resistance

TABLE 6: Different grid size levels in 3D cavity.

Grid size level	Number of cells
Finest grid	125,000
Coarse grid level 1	30,698
Coarse grid level 2	8324
Coarse grid level 3	2706

FIGURE 22: Residual reduction for the second time step.

TABLE 7: Convergence features for the 3D cavity in $Re = 1000$ for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	3651	1	~179,000
3-grid V cycle	2339	1.56	~230,000
3-grid W cycle	2131	1.71	~230,000
4-grid V cycle	2258	1.62	~235,000
4-grid W cycle	1981	1.85	~235,000

TABLE 8: Convergence features for the TSNFL in $Re = 500$ for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	1756	1	~256,000
3-grid V cycle	1162	1.51	~351,000
3-grid W cycle	1115	1.57	~351,000
4-grid V cycle	1069	1.64	~372,000
4-grid W cycle	975	1.80	~372,000

becomes steady. These oscillations appear due to the heave and pitch motion of the barge. According to Figure 36, it takes about 5 seconds for barge motions to become stable.

The computed resistance is compared with experimental data in Table 15. The result shows that the presented numerical solver can estimate barge resistance with good accuracy.

The free surface near the barge is also compared with the experimental photo in Figure 37.

7.6. Resistance and Motion of a High-Speed Catamaran. As seen in the previous section, a low-speed barge has low heave and pitch motion. Therefore, in order to examine the

capability of the proposed method to solve ship motions, it is recommended to investigate a high-speed craft. For this reason, in this section, a high-speed catamaran is investigated at three different (constant) velocities. Table 16 shows the main characteristic of the catamaran.

A total of 180,992 structural cells are used to simulate the catamaran at three constant velocities ($Vel = 5, 16.5, 24$) (Figure 38).

Like the previous test case, the catamaran moves with a constant velocity while it is allowed to exhibit heave and pitch motions (2-DOF). Each simulation is performed until three main hydrodynamic parameters of catamaran become steady: heave motion, trim angle, and resistance force. Figure 39 shows the catamaran in steady conditions at different speeds.

Heave, trim angle, and resistance force are compared with the work of Panahi et al. [28] in Figure 40.

As shown in Figure 40, the presented model could predict the heave and pitch motion with reasonable accuracy. The resistance which is the result of these two parameters is also well calculated. For this test case, the multigrid method is applied using the grids which are listed in Table 17.

The residual reduction for the second time step is plotted in Figure 41.

Like other simulations, all types of cycles for 50 time steps are applied to investigate their performance Table 18. Based on Table 18, W cycle with a four-grid level was chosen for all simulations.

7.7. Resistance and Motion of Oceanography Research Vessel. The final test case involves the calculation of the resistance and motion of an Oceanography Research Vessel (RV) in still water [45]. Table 19 shows the main characteristic of the vessel.

The resistance of the vessel is estimated experimentally at the Vienna Model Basin [45]. The wooden model of 50 m RV is made in 1:12 scale. To achieve better accuracy, two shaft A-brackets and two rudders are applied to the model as shown in Figure 42.

Based on the towing tank test, the vessel resistance at different speeds is estimated and presented in Table 20.

Based on the geometrical full-scale model, a fully unstructured grid is used for numerical simulation. The discretized RV is shown in Figure 43.

It should also be noted that A-brackets and rudders are modeled in a numerical model like an experimental wooden model for better accuracy Figure 44. These appendages play an important role in creating hydrodynamic resistance of the ship hull.

Based on the proposed prolongation algorithm, three different grid size levels are created in numerical simulations. Table 21 shows the grid size properties. It should be noted that the numerical solver could not solve the coarse grid level 3 because of details in the appendage part of RV. In this part, the coarse cells become numerically unstable. Therefore, in this test case, three grid levels are used in the multigrid cycles.

FIGURE 23: The geometric model of two-sided nonfacing lid-driven cubical cavity [43].

FIGURE 24: Velocity components (u and w) at the centerline.

FIGURE 25: Velocity component contours of u (right side: present study; left side: [43]).

FIGURE 26: Velocity component contours of w (right side: present study; left side: [43]).

TABLE 9: Different grid size levels in TSNFL.

Grid size level	Number of cells
Finest grid	59,319
Coarse grid level 1	14,568
Coarse grid level 2	3950
Coarse grid level 3	1284

FIGURE 27: Residual reduction for the second time step.

FIGURE 28: Continued.

FIGURE 28: Geometry and initial condition of the problem.

TABLE 10: Different grid size levels in the collapse of a water column with an obstacle.

Grid size level	Number of cells
Finest grid	198,430
Coarse grid level 1	49,851
Coarse grid level 2	13,125
Coarse grid level 3	3454

FIGURE 29: Residual reduction for collapse of water column with obstacle in the second time step.

TABLE 11: Convergence features for collapse of water column with obstacle for 100 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	5933	1	~765,000
3-grid V	4238	1.40	~101,000
3-grid W	3731	1.59	~101,000
4-grid V cycle	3640	1.63	~1126,000
4-grid W cycle	3278	1.81	~1126,000

As observed, in fully unstructured grids, the proposed coarsening algorithm has a better performance. Comparison of the coarsening in an unstructured triangular cavity test case with the structured BFS or dam break grids reveals that, in unstructured grids, the coarsening ratio is about 4.5, while in structured grids, the ratio is almost 4. This is due to the fact that, in structured grids, the number of sharing cells for each vertex is less than the unstructured grids. For example, in the triangular cavity, there are vertexes with 6

sharing cells, while this number is 4 in the BFS test case. On the other hand, it may have a coarsening limitation on unstructured grids. For example in this case, “coarse grid level 3” could not be created because, by applying the presented coarsening algorithm, a very complex cell would be generated which occupies the whole space between rudders and brackets.

In the forward motion of the ships, heave and pitch motions are dominantly related to four other motions. In

FIGURE 30: Water height in the reservoir at H1 and H2. (a) Free surface elevation at point H1. (b) Free surface elevation at point H2.

FIGURE 31: Pressure value at (a) P1, (b) P2, and (c) P3.

FIGURE 32: Continued.

FIGURE 32: Qualitative comparison of free surface with experimental images at time (a) 0.4 s and (b) 0.56 s.

FIGURE 33: Barge geometry.

FIGURE 35: Residual reduction for 2-DOF motion barge in the second time step.

TABLE 12: Barge characteristics [26].

Length (L)	1.05 (m)
Breadth (B)	0.29 (m)
Depth (D)	0.025 (m)
Draft (d)	1.0 (m)
Mass	7.26 (kg)
I_{YY}	0.7 (kgm^2)
KG	0.025 (m)

TABLE 14: Convergence features for barge resistance for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	18,717	1	~706,000
3-grid V cycle	12,820	1.46	~910,000
3-grid W cycle	11,922	1.57	~910,000
4-grid V cycle	11,625	1.61	~982,000
4-grid W cycle	10,009	1.87	~982,000

FIGURE 34: Barge structural mesh (158,891 cells). Three grid level sizes are used in multigrid solver which are listed in Table 13.

TABLE 13: Different grid size levels in the barge.

Grid size level	Number of cells
Finest grid	158,891
Coarse grid level 1	39,872
Coarse grid level 2	10,438
Coarse grid level 3	2973

FIGURE 36: Resistance of the barge in calm water.

TABLE 15: Comparison of barge resistance with experimental data.

Method	Resistance (N)	Error (%)
Experimental [2]	3.53	—
Numerical simulation	3.41	3.4

FIGURE 37: Comparison of free surface deformation in front of the barge with the experimental snapshot.

TABLE 16: Catamaran characteristics [28].

Length (m)	12.3
Breadth (m)	4.6
Draught (m)	0.95
Mass (ton)	17.850
Vertical center of gravity (m)	0.45
Longitudinal center of gravity (ton)	3.81
Block coefficient	0.33

FIGURE 38: High-speed craft catamaran structural mesh (180,992 cells).

FIGURE 39: Catamaran in a steady condition at different speeds. (a) $V = 5$ m/s, (b) $V = 16.5$ m/s, (c) $V = 24$ m/s, and (d) $V = 24$ m/s.

FIGURE 40: Heave, trim angle, and resistance of high-speed craft catamaran.

TABLE 17: Different grid size levels for the catamaran.

Grid size level	Number of cells
Finest grid	180,992
Coarse grid level 1	45,401
Coarse grid level 2	11,820
Coarse grid level 3	3369

FIGURE 41: Residual reduction for a catamaran in the second time step.

TABLE 18: Convergence features for barge resistance for 50 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	805,301	1	36,244
3-grid V cycle	1,036,200	1.55	23,384
3-grid W cycle	1,036,200	1.76	20,592
4-grid V cycle	1,119,107	1.83	19,806
4-grid W cycle	1,119,107	2.01	18,032

TABLE 19: Main characteristics of RV.

Length between perpendiculars (LBP) (m)	44.70
Breadth (m)	10.00
Draught at midship (m)	3.30
Draught at forward perpendicular (m)	3.30
Draught at aft perpendicular (m)	3.30
Displacement mass (ton)	896
Block coefficient	0.5923
Maximum speed (knot)	14

FIGURE 42: Wooden model of RV50 vessel.

TABLE 20: RV50 resistance at different speeds.

Vessel speed (knot)	Total resistance (kN)
3.00	2.00
4.00	5.10
5.00	8.10
6.00	11.40
7.00	15.60
8.00	21.40
9.00	29.00
10.00	38.70
11.00	50.70
12.00	65.00
13.00	82.00

FIGURE 44: A-brackets and rudders in a numerical model.

FIGURE 43: Fully unstructured mesh of RV50.

TABLE 21: Different grid size levels for RV50 simulation.

Grid size level	Number of cells
Finest grid	200,455
Coarse grid level 1	44,545
Coarse grid level 2	10,606

FIGURE 45: Resistance curves for different RV velocities.

FIGURE 46: Comparison of resistance curves from the numerical results and experimental data [45].

FIGURE 47: Heave motion curves at different velocities.

this case, two degrees of freedom of RV50 (heave and pitch motions) are determined from hydrodynamic forces. Simulation is performed for different velocities until the resistance force becomes nearly steady. Figure 45 shows the resistance curves for different RV velocities.

Figure 46 compares the simulated resistance curve with experimental data [45]. The comparison shows good agreement and indicates that the proposed numerical simulation can predict ship resistance with reasonable accuracy.

In displacement ships, with rather a low operative speed, the heave and pitch motions are small. In this case, the pitch

angle is almost zero and the heave motion is ignorable. The heave diagram is plotted in Figure 47 for different velocities.

In order to capture free surface elevation with good resolution, the grid sizes are chosen to be finer near the ship and free surface related to far field. Figure 48 shows a free surface near the ship hull at different velocities.

As all the simulations are performed by the proposed multigrid algorithm, the residual reduction curve can be investigated in order to choose the best multigrid cycle. Figure 49 shows the residual curves for different multigrid cycles in the second time step.

Table 22 displays the CPU time and memory allocations of different iterative methods for 100 time steps.

FIGURE 48: Free surface near the ship hull at different velocities.

FIGURE 49: Residual reduction for the second time step.

TABLE 22: Convergence features for RV50 resistance in 100 time steps.

The method	CPU time (s)	CPU speed-up ratio	Memory allocation (KB)
Single-grid	3173	1	~729,088
3-grid V cycle	1132	2.8	~1,312,358
3-grid W cycle	881	3.6	~1,837,302

As observed in Table 22, the W cycle is proved to be more effective in terms of convergence and CPU time in the simulation of the research vehicle's resistance.

8. Conclusions

An unsteady 3D Navier-Stokes solver combined with the VOF approach is developed to simulate different fluid phenomena with emphasis on the ship motion. Applying a fully unstructured grid to the solver makes it possible to study the ship motions with all hull appendages such as rudder and bracket. To speed up the convergence rate, an agglomeration multigrid method is used with a new geometrical agglomeration technique to create coarse grids. For this purpose, a fully nonstandard unstructured grid with multiple faces is applied to the solver. The new data structure makes it possible to increase the accuracy in coarsened grids, especially in ship's hull details like appendages. As the fractional step method is the technique for coupling velocity and pressure field for solving Navier-Stokes equations, the multigrid method is applied to pressure Poisson equations. This combination with new nonstandard grids is one of the main novelties of the current article. A rather simple prolongation/restriction operator is used to transfer data between coarse/fine grids in both V and W cycles. In order to validate the accuracy and investigate the performance of the proposed method, two distinguishing parameters are studied: computational time and residual reduction. In each test case, these two parameters are compared to ascertain which cycle has better performance.

It is determined that W cycle reduces the residual better than V cycle. This is due to the fact that, in one cycle of W scheme, double data transferring to coarse grids appear. This action helps the solver to damp the lower frequencies, easily. However, this data transferring takes more CPU time compared to V scheme. Therefore, one can see in the test cases that W cycle is more suitable for more complicated problems with a large number of grids. Consequently, V cycle is preferred to be used in simple cases like two-dimensional problems, but the most important issue is that the multigrid method can speed up the solver up to 3 times without reducing accuracy.

Data Availability

Data will be made available upon request.

Conflicts of Interest

The authors further declare that there are no conflicts of interest

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] Y. Zang and R. L. Street, "A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains," *International Journal for Numerical Methods in Fluids*, vol. 20, no. 5, pp. 341–361, 1995.
- [2] J. C. Meza and R. S. Tuminaro, "A multigrid preconditioner for the semiconductor equations," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 118–132, 1996.
- [3] P. T. Lin, "Implicit time dependent calculations for compressible and incompressible flows on unstructured meshes," M. Sc. Thesis, Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA, 1994.
- [4] C. Liu, X. Zheng, and C. H. Sung, "Preconditioned multigrid methods for unsteady incompressible flows," *Journal of Computational Physics*, vol. 139, no. 1, pp. 35–57, 1998.
- [5] I. Ginzburg and G. Wittum, "Two-phase flows on Interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants," *Journal of Computational Physics*, vol. 166, no. 2, pp. 302–335, 2001.
- [6] H. S. Tang, S. Casey Jones, and F. Sotiropoulos, "An overset-grid method for 3D unsteady incompressible flows," *Journal of Computational Physics*, vol. 191, no. 2, pp. 567–600, 2003.
- [7] C. H. Tai and Y. Zhao, "Parallel unsteady incompressible viscous flow computations using an unstructured multigrid method," *Journal of Computational Physics*, vol. 192, no. 1, pp. 277–311, 2003.
- [8] C. H. Tai and Y. Zhao, "A finite volume unstructured multigrid method for efficient computation of unsteady incompressible viscous flows," *International Journal for Numerical Methods in Fluids*, vol. 46, no. 1, pp. 59–84, 2004.
- [9] N. K. Lambropoulos, D. G. Koubogiannis, and K. C. Giannakoglou, "Acceleration of a Navier-Stokes equation solver for unstructured grids using agglomeration multigrid and parallel processing," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 9–11, pp. 781–803, 2004.
- [10] J. Li, B. Yu, Y. Zhao, Y. Wang, and W. Li, "Flux conservation principle on construction of residual restriction operators for multigrid method," *International Communications in Heat and Mass Transfer*, vol. 54, pp. 60–66, 2014.
- [11] X. Lv, Y. Zhao, X. Y. Huang, G. H. Xia, and Z. J. Wang, "An efficient parallel/unstructured-multigrid preconditioned implicit method for simulating 3D unsteady compressible flows with moving objects," *Journal of Computational Physics*, vol. 215, no. 2, pp. 661–690, 2006.
- [12] N. B. Cheikh, B. B. Beya, and T. Lili, "A multigrid method for solving the Navier-Stokes/Boussinesq equations," *Communications in Numerical Methods in Engineering*, vol. 24, no. 8, pp. 671–681, 2007.
- [13] B. E. Griffith, "An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7565–7595, 2009.
- [14] K. D. Santhosh, D. K. Anoop, and D. Anupam, "A multigrid-accelerated three-dimensional transient flow code and its application to a," *New Test Problem*, vol. 22, pp. 838–846, 2010.

- [15] L. Zong-zhe, W. Zheng-hua, C. Wei, and Y. Lu, "A strong coupled agglomeration multigrid for Navier-Stokes equations on 3D Hybrid Grids," *Procedia Engineering*, vol. 15, pp. 3916–3920, 2011.
- [16] O. San and A. E. Staples, "A coarse-grid projection method for accelerating incompressible flow computations," *Journal of Computational Physics*, vol. 233, pp. 480–508, 2013.
- [17] S. Langer, "Agglomeration multigrid methods with implicit Runge-Kutta smoothers applied to aerodynamic simulations on unstructured grids," *Journal of Computational Physics*, vol. 277, no. 15, pp. 72–100, 2014.
- [18] W. Cheng and R. Samtaney, "A high-resolution code for large eddy simulation of incompressible turbulent boundary layer flows," *Computers & Fluids*, vol. 92, pp. 82–92, 2014.
- [19] J. Stiller, "Robust multigrid for high-order discontinuous Galerkin methods: a fast Poisson solver suitable for high-aspect ratio Cartesian grids," *Journal of Computational Physics*, vol. 327, pp. 317–336, 2016.
- [20] G. N. Lygidakis, S. S. Sarakinos, and I. K. Nikolos, "Comparison of different agglomeration multigrid schemes for compressible and incompressible flow simulations," *Advances in Engineering Software*, vol. 101, pp. 77–97, 2016.
- [21] E. N. Matthioudakis, V. G. Mandikas, G. V. Kozyrakas, N. A. Kampanis, and J. A. Ekaterinaris, "Multigrid cell-centered techniques for high-order incompressible flow numerical solutions," *Aerospace Science and Technology*, vol. 64, pp. 85–101, 2017.
- [22] S. Abide and B. Zeghamati, "Multigrid defect correction and fourth-order compact scheme for Poisson's equation," *Computers & Mathematics with Applications*, vol. 73, no. 7, pp. 1433–1444, 2017.
- [23] Y. Liu, W. Zhang, and J. Kou, "Mode multigrid - a novel convergence acceleration method," *Aerospace Science and Technology*, vol. 92, pp. 605–619, 2019.
- [24] S. T. Ha, H. Y. Yoon, and H. Choi, "Performance comparison of interpolation operators of multigrid methods based on distance weight and area (volume) intersection approaches part I: finite volume discretization," *Journal of Mechanical Science and Technology*, vol. 33, no. 5, 2019.
- [25] D. Kim and H. Choi, "A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids," *Journal of Computational Physics*, vol. 162, no. 2, pp. 411–428, 2000.
- [26] R. Panahi, E. Jahanbakhsh, and M. S. Seif, "Development of a VOF-fractional step solver for floating body motion simulation," *Applied Ocean Research*, vol. 28, no. 3, pp. 171–181, 2006.
- [27] O. Ubbink and R. I. Issa, "A method for capturing sharp fluid interfaces on arbitrary meshes," *Journal of Computational Physics*, vol. 153, no. 1, pp. 26–50, 1999.
- [28] R. Panahi, E. Jahanbakhsh, and M. S. Seif, "Towards simulation of 3D nonlinear high-speed vessels motion," *Ocean Engineering*, vol. 36, no. 3-4, pp. 256–265, 2009.
- [29] W. A. Smith, "Multigrid solution of transonic flow on unstructured grids," in *Proceedings of the ASME Annual Meeting Recent Advances and Applications in Computational Fluid Dynamics*, Washington, DC, USA, 1990.
- [30] M.-H. Lallemand, H. Steve, and A. Dervieux, "Unstructured multigridding by volume agglomeration: current status," *Computers & Fluids*, vol. 21, no. 3, pp. 397–433, 1992.
- [31] H. Jasak, *Error analysis and estimation for finite volume method with application to fluid flows*, Ph.D. Thesis, University of London, London, UK, 1996.
- [32] B. P. Leonard, "The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection," *Computer Methods in Applied Mechanics and Engineering*, vol. 88, no. 1, pp. 17–74, 1991.
- [33] H. Gaskell and A. K. C. Lau, "Curvature-compensated convective transport: SMART, a new Boundedness-preserving transport algorithm," *International Journal of Numerical Method Fluids*, vol. 8, pp. 617–641, 1998.
- [34] N. Okamoto, K. Nakahashi, and S. Obayashi, "A coarse generation algorithm for agglomeration multigrid method on unstructured grids," in *Proceedings of the 36th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, January 1998.
- [35] W. Briggs, V. Henson, and S. McCormick, "A multigrid tutorial," SIAM, Philadelphia, PA, USA, 2000.
- [36] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, Cambridge, MA, USA, 2001.
- [37] J. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, Germany, 1996.
- [38] A. Jagannathan, M. Ranjith, and D. Manhar, "A spectral method for the triangular cavity flow," *Computers and Fluids*, vol. 95, pp. 40–48, 2014.
- [39] D. K. Gartling, "A test problem for outflow boundary conditions-flow over a backward-facing step," *International Journal for Numerical Methods in Fluids*, vol. 11, no. 7, pp. 953–967, 1990.
- [40] E. Erturk, "Numerical solutions of 2-D steady incompressible flow over a backward-facing step, Part I: high Reynolds number solutions," *Computers & Fluids*, vol. 37, no. 6, pp. 633–655, 2008.
- [41] B.-n. Jiang, T. L. Lin, and L. A. Povinelli, "Large-scale computation of incompressible viscous flow by least-squares finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 114, no. 3-4, pp. 213–231, 1994.
- [42] H. C. Ku, R. S. Hirsh, and T. D. Taylor, "A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 70, no. 2, pp. 439–462, 1987.
- [43] B. Ben Beya and T. Lili, "Three-dimensional incompressible flow in a two-sided non-facing lid-driven cubical cavity," *Comptes Rendus Mécanique*, vol. 336, no. 11-12, pp. 863–872, 2008.
- [44] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, and B. Buchner, "A volume-of-fluid based simulation method for wave impact problems," *Journal of Computational Physics*, vol. 206, no. 1, pp. 363–393, 2005.
- [45] Test Report No. 2512/01, Model No.2512, Vienna Model Basin, Italy, 2011.