Implementation of the Ghost Fluid Method for Free Surface Flows in Polyhedral Finite Volume Framework

Vuko Vukčević[a,*], Hrvoje Jasak[a,b], Inno Gatin[a]

[a] *University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Ivana Lučića 5, Zagreb, Croatia*
[b] *Wikki Ltd, 459 Southbank House, SE1 7SJ, London, United Kingdom*

Abstract

This paper presents an extension of the Ghost Fluid Method to arbitrary polyhedral Finite Volume framework for free surface flow simulations, primarily intended for marine hydrodynamics applications. Two immiscible, incompressible fluids are implicitly coupled via interface jump conditions, allowing the formulation of a single set of equations for both fluids. The jump conditions at the free surface are discretised with one sided extrapolates, using a compact computational stencil in second–order accurate, collocated polyhedral Finite Volume Method. The free surface is captured using the Volume–of–Fluid method with an additional compressive term. Even though the Volume–of–Fluid method is used, density and dynamic pressure exhibit sharp distribution at the interface due to jump conditions. The paper also demonstrates how the conditionally averaged equations with segregated solution algorithms cause spurious velocities at the free surface, which are resolved by the present approach since the Ghost Fluid Method relocates the pressure–density coupling inside the pressure equation. The method is implemented in OpenFOAM Computational Continuum Mechanics software, and the verification and validation is performed on two sets of test cases: free surface flow over a ramp and a steady resistance simulation of a container ship free to sink and trim.

Keywords:

Ghost Fluid Method, Free surface flow, Volume–of–Fluid, Collocated polyhedral Finite Volume method, OpenFOAM, Marine hydrodynamics, Validation and verification, Steady resistance with sinkage and trim

1. Introduction

With immense increase in computer resources over the past decades, multiphase flow modelling has become common practice. The iterative nature of Computational Fluid Dynamics (CFD) makes it suitable for handling non–linear and complex equations that govern multiphase flows.

Water–air free surface flows cover wide range of phenomena, from small–scale flows [1, 2, 3] to large–scale marine hydrodynamic applications [4, 5, 6, 7]. In

*Corresponding author.
Email addresses: vuko.vukcevic@fsb.hr (Vuko Vukčević), hrvoje.jasak@fsb.hr (Hrvoje Jasak), h.jasak@wikki.co.uk (Hrvoje Jasak), innogatin@gmail.com (Inno Gatin)

this work, we present a general Ghost Fluid Method (GFM) [8, 9, 10, 11, 12, 13, 14] model for incompressible, multiphase flow in the arbitrary polyhedral Finite Volume (FV) framework. The method is primarily intended for large length–scale applications related to marine hydrodynamics [15].

Free–surface flows are often modelled with [16]: Volume–of–Fluid (VOF) [17, 18, 19, 20], Lagrangian tracking and Level Set (LS) methods [21, 22, 23, 24]. Being conservative, the VOF method is well-established and extensively used in CFD regarding non–linear surface wave propagation [25, 26, 27, 28, 29]. However, special procedures need to be employed to ensure boundedness of the solution and to confine smearing of the interface [17, 19]. The LS method based on the signed distance function [30] is not conservative and often requires redistancing algorithms [31, 32]. On the other hand, LS method based on hyperbolic tangent profile is conservative [23, 24] with additional terms that compress the interface and maintain the specified profile. The Phase Field method [33, 34] also represents a variant of the interface capturing approach where the tangent hyperbolic profile is transported, while the preservation of the profile is achieved during the solution process. Another approach is sharp interface tracking method using moving grids [1, 2], which introduces additional grid deformation equations that increase the CPU time. Complex geometries (e.g. a ship hull) and complex flow characteristics (e.g. wave breaking) become challenging to simulate with this approach.

1.1. Related studies

Two approaches for the treatment of jump conditions at the free surface exist: embedded free surface method [35] and the GFM [11]. Both approaches consider two fluids with a sharp interface, the only difference being the treatment (discretisation) of interface jump conditions, as indicated by Wang *et al.* [35].

Johansen and Colella [36] developed the embedded boundary method for Poisson's equation, while Crockett *et al.* [37] embedded the discontinuous jump conditions in Poisson and heat equations, achieving second–order accuracy. Recently, Wang *et al.* [35] extended the method for two phase incompressible flows. The publications regarding the embedding technique considered structured Cartesian grids.

The GFM for jump condition treatment across a sharp, moving interface, represents an active area of research in the past two decades. Fedkiw *et al.* [8] developed the GFM coupled with the LS interface capturing to efficiently capture discontinuities in deflagration and detonation problems. The method has been extended by Kang *et al.* [9] to simulate multiphase incompressible and laminar flow. Desjardins *et al.* [10] presented the methodology based on the GFM with the conservative LS method developed by Olsson and Kreiss [23], and remarked that the GFM provides a good framework for two phase flows with large density variations. Recently, Lalanne *et al.* [14] provided an extensive overview of the treatment of discontinuous viscosity in two phase flow utilising the GFM with LS interface capturing. The GFM has been also used

alongside the VOF method for compressible multiphase flows by Bo and Grove [12]. The above mentioned publications on the GFM utilise structured Cartesian grids.

Huang *et al.* [11] implemented the GFM with the LS interface capturing in the finite difference framework on curvilinear grids. The jump conditions are discretised using second–order accurate schemes and the LS method is used to calculate one–sided extrapolates of pressure. The authors note that the method is suitable for large length–scale free surface water–air flows encountered in marine hydrodynamics, although they report that the resulting matrix from pressure equation is stiff. Also, it is not clear from the publication how they treat density and pressure discontinuity in the momentum equation.

1.2. Present approach

In this paper, we present the extension of the GFM implemented by Huang *et al.* [11] in arbitrary polyhedral FV framework with collocated arrangement [38]. The methodology has not been implemented to our present knowledge, although Queutey and Visonneau [39] present a similar approach where the free surface is assumed to be aligned with the grid faces. Following Huang *et al.* [11], special interpolation schemes for density and pressure are derived and presented, where it is shown that the resulting pressure equation matrix remains symmetric. Due to discretisation of pressure jump conditions in the GFM, the smearing of the interface only affects viscous and turbulent effects, whereas density and pressure are calculated assuming a sharp interface. Therefore, the VOF method [19] is used in present work. Two–equation turbulence modelling [40] is used, while the surface tension effects are neglected in present work since they are of minor importance for marine hydrodynamic flows. The GFM treatment of surface tension shall be the focus of future work.

The method is implemented in foam–extend–3.2, a scientific community driven fork of the OpenFOAM CFD software [41]. Support for parallel computations is achieved with domain decomposition and efficient linear system solvers. Pressure–velocity–VOF coupling is resolved in a segregated manner using a combination of SIMPLE [42] and PISO [43] algorithms. Validation of the model is performed on two sets of test cases: free surface flow over a ramp and steady resistance simulations of the Kriso (Moeri) Container Ship (KCS) with dynamic sinkage and trim, where the results are compared to analytical and experimental data, respectively. Verification is achieved by performing grid and iterative uncertainty assessment.

The paper is organised as follows. The cause of spurious air velocities is briefly discussed in Sec. 1.3. General two–phase flow model is presented in Sec. 2 with approximate jump conditions at the interface that are valid for large–scale flows. Numerical model is presented in Sec. 3 with emphasis on the discretisation of dynamic pressure terms appearing in the momentum and pressure equation. Derivation of interface–corrected interpolation schemes for the dynamic pres-

sure and density using the free surface jump conditions is presented in detail in Sec. 4. Solution algorithm is briefly described in Sec. 5 and the verification and validation of the presented method is presented in Sec. 6.

1.3. The cause of spurious air velocities

Incompressible, Newtonian flow is governed by Navier–Stokes equations. Multiple incompressible phases are coupled with appropriate jump conditions at the interface [44], where the kinematic boundary condition ensures continuity of velocity field across the interface, while tangential and normal stress balance yield jump conditions for velocity gradient and pressure. Some researchers [25, 26, 27, 28, 29] in offshore and marine CFD use a two–phase momentum equation that is derived based on conditional averaging [45] procedure which includes variable two–phase density. The two–phase momentum equation has the following form:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \nabla\cdot\left(\mu_{eff}\nabla\mathbf{u}\right) = -\nabla p_d - \mathbf{g}\cdot\mathbf{x}\nabla\rho + \\ + \nabla\mathbf{u}\cdot\nabla\mu_{eff} + \sigma\kappa\nabla\alpha\,, \tag{1}$$

where ρ is the two–phase density field, $\mathbf{u}$ is the velocity field, μ_e is the two–phase effective dynamic viscosity, p_d is the dynamic pressure, $\mathbf{g}$ is the gravitational acceleration vector, $\mathbf{x}$ is the position vector, σ is the surface tension coefficient, κ is the mean curvature of the interface and α is the volume fraction using VOF approach. Dynamic pressure gradient ∇p_d and density gradient (present near the interface) $\nabla\rho$ should be balanced in a hydrostatic test case. Hence, the coupling between dynamic pressure and density is resolved in the momentum equation, which leads to spurious air velocities near the interface.

Consider a hydrostatic case of inviscid fluid where the surface tension is neglected. Eqn. (1) becomes:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} = -\nabla p_d - \mathbf{g}\cdot\mathbf{x}\nabla\rho = \mathbf{S_u}\,, \tag{2}$$

where the source term $\mathbf{S_u}$ represents imbalance between dynamic pressure gradient and density gradient, which is present during segregated numerical solution as used by many CFD algorithms [25, 26, 27, 28, 29, 11, 6, 39]. The source term causes temporal change in the velocity field. The erroneous change in the velocity field is higher in the lighter phase because of ρ pre–factor in the time derivative term. Furthermore, dynamic pressure gradient and density gradient are often discretised with usual gradient discretisation schemes that do not take the density discontinuity into account, making them inefficient near the interface. It is important to note that this consideration does not include surface tension effects, thus, this observation is unrelated to parasitic currents due to numerical modelling issues related to the surface tension in atomisation calculations using the Continuous Surface Stress (CSS) model [46]. The model implementing the conditionally averaged equations outlined above is compared to the present model for the hydrostatic test case in Sec. 6.

2. Two–phase flow model

This section presents the mathematical model for incompressible, turbulent, multiphase flow of two immiscible fluids with a sharp interface. The appropriate jump conditions at the interface are first discussed, followed by the incompressible, Newtonian fluid equations governing the flow. Turbulence is accounted for in a generic way, allowing the use of general turbulence models [47].

2.1. Free surface jump conditions

The interface between two fluids Γ separates the two phases, here water and air, without loss of generality. In each fluid, density is assumed constant, *i.e.* $\rho = \rho_w$ in water and $\rho = \rho_a$ in air. Following notation used by GFM authors [11, 10], the jump of density across the free surface can be written as:

$$[\rho] = \rho^- - \rho^+ , \tag{3}$$

where superscripts + and − denote heavier and lighter fluid, respectively.

To facilitate the implementation of the GFM in arbitrary polyhedral FV framework, effective kinematic viscosity is assumed continuous across the free surface using the volume fraction, α from the VOF approach (see Sec. 2.3):

$$\nu_e = \alpha \nu_{e,w} + (1 - \alpha)\nu_{e,a} , \tag{4}$$

where $\nu_{e,w}$ is the effective kinematic viscosity in water and $\nu_{e,a}$ is the effective kinematic viscosity in air, allowing the use of general eddy–viscosity turbulence models [47]. Eqn. (4) states that the effective kinematic viscosity does not have a jump across the free surface, albeit it may have a steep gradient if the smearing of the interface in the VOF approach is confined to a narrow region. Similar definition has been used by Olsson and Kreiss [23], who used smeared Heaviside function instead of α, and Desjardins *et al.* [10], who used a tangent hyperbolic LS profile. This approximation is justified for large–scale flows where viscosity effects have minor influence on the interface behaviour [10]. A more advanced treatment of the viscosity jump conditions has been recently presented by Lalanne *et al.* [14], who also gave an excellent overview of other possible approaches regarding numerical treatment of viscosity.

The kinematic boundary condition [44, 48] defines a continuous velocity field across the free surface:

$$[\mathbf{u}] = \mathbf{u}^- - \mathbf{u}^+ = 0 . \tag{5}$$

Eqn. (5) states that the velocity field infinitesimally close to the free surface in the lighter fluid must be the same as the velocity field infinitesimally close to the free surface in the heavier fluid.

Furthermore, neglecting the surface tension effects as in [39, 11] yields a continuous pressure field:

$$[p] = 0 . \tag{6}$$

Eqn. (3), Eqn. (5) and Eqn. (6) present jump conditions that need to be taken into account to numerically model the free surface flow.

2.2. *Incompressible two–phase flow*

As both fluids are assumed incompressible with piece–wise constant density and the velocity field is continuous, the continuity constraint reads:

$$\nabla\cdot\mathbf{u} = 0\,, \tag{7}$$

Following Desjardins *et al.* [10], the momentum equation for an incompressible Newtonian fluid in a gravitational field reads:

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{uu}) - \nabla\cdot(\nu_e\nabla\mathbf{u}) = -\frac{1}{\rho}\nabla p + \mathbf{g}\,, \tag{8}$$

where ν_e is the effective kinematic viscosity defined by Eqn. (4), p is the pressure field and $\mathbf{g}$ is the gravitational acceleration. Since incompressible flow is assumed, ρ denotes piece–wise constant density field. The right hand side (RHS) of Eqn. (8) can be written in a more convenient way:

$$-\frac{1}{\rho}\nabla p + \mathbf{g} = -\beta\nabla p + \mathbf{g}\,, \tag{9}$$

introducing:

$$\beta = \frac{1}{\rho}\,. \tag{10}$$

Two terms in Eqn. (8) denoting the pressure gradient and gravitational acceleration can be written as follows:

$$\begin{aligned} -\beta\nabla p + \mathbf{g} = -\beta\nabla p + \nabla(\mathbf{g}\cdot\mathbf{x}) &= -\beta\nabla\left(p - \frac{\mathbf{g}\cdot\mathbf{x}}{\beta}\right) \\ &= -\beta\nabla p_d\,. \end{aligned} \tag{11}$$

It should be noted that in the second identity, the inverse density may be positioned inside the gradient operator as the two phases are considered incompressible and the density jump condition shall be treated via GFM in Sec. 4. The decomposition of pressure into hydrostatic and dynamic part reads:

$$p = p_d + \frac{\mathbf{g}\cdot\mathbf{x}}{\beta}\,. \tag{12}$$

The momentum equation for incompressible, turbulent, two–phase flow, Eqn. (8) has the final form:

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{uu}) - \nabla\cdot(\nu_e\nabla\mathbf{u}) = -\beta\nabla p_d\,. \tag{13}$$

Compared to Eqn. (1), $\nabla\rho$ term is absent in Eqn. (13) due to assumed piece–wise constant density of the fluid that shall be treated in Sec. 4.

Using decomposition of pressure given by Eqn. (12), the pressure jump condition defined in Eqn. (6) is rewritten in terms of dynamic pressure:

$$[p_d] = -\left[\frac{1}{\beta}\right]\mathbf{g}\cdot\mathbf{x}\,. \tag{14}$$

Furthermore, due to continuous velocity field and the assumed continuity of the kinematic viscosity, the dynamic pressure gradient term appearing on the right–hand–side (RHS) of Eqn. (13) is also continuous:

$$[\beta\nabla p_d] = 0\,, \tag{15}$$

which represents an additional jump condition that needs to be taken into account, as indicated by a number of authors [10, 11, 39].

2.3. Interface capturing

Capturing of the free surface between two phases is achieved with the VOF method [17]. The VOF method is based on the indicator function α, which represents the volume fraction:

$$\alpha = \frac{V_w}{V}\,, \tag{16}$$

where V_w is the volume of water inside a control volume V. VOF method is conservative because α represents a physical, conserved property, bounded between 0 and 1. Following Rusche [19], transport equation for α reads:

$$\frac{\partial \alpha}{\partial t} + \nabla\cdot(\mathbf{u}\alpha) + \nabla\cdot(\mathbf{u}_r\alpha(1-\alpha)) = 0\,, \tag{17}$$

where the last term serves to prevent excessive smearing of the free surface based on the compressive velocity field $\mathbf{u}_r$, which shall be defined in Sec. 3.4. The term is active only in the smeared interface region due to the $\alpha(1-\alpha)$ non–linear pre–factor. The VOF method with the additional compressive term is similar to the conservative LS method proposed by Olsson and Kreiss [23, 24]. The methods differ in the definition of the indicator field α and in the additional diffusion term present in the conservative LS method to stabilise the advection.

Once calculated, α is used to calculate the effective viscosity using Eqn. (4), making it a continuous, albeit steep function. We stress that ρ, and consequently β (see Eqn. (10)) field are not smeared using α: only ρ_w, ρ_a and the location of the interface given by $\alpha = 0.5$ will be used in the discretisation of the momentum equation, Eqn. (13), obeying dynamic pressure jump conditions given by Eqn. (14) and Eqn. (15).

The model can be easily extended for more than two phases by including additional VOF equations to capture multiple phases, as demonstrated by Kissling *et al.* [20].

3. Numerical model

Governing partial differential equations, Eqn. (7), (8), (17) are discretised in space using a second–order accurate, collocated FV method for arbitrary polyhedral (unstructured) grids [38]. A polyhedral control volume (CV), or cell, presented in Figure 1, has a number of neighbours connected through common faces, where $\mathbf{s}_f$ represents surface area vector and $\mathbf{d}_f$ is the distance vector from cell centre P to cell centre N.

3.1. Finite Volume discretisation

We shall divide the analysis of the FV discretisation into two parts. We shall present the general FV method with arbitrary discretisation schemes. Without

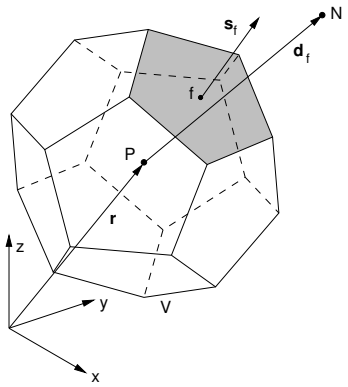


Figure 1: Polyhedral control volume. Control volume P shares a common face with its immediate neighbour N.

detailed analysis, terms enclosed in curly braces $\{\cdot\}$ (as opposed to $[\cdot]$ as used by Rusche [19]) represent implicit FV discretisation, while other terms represent explicit FV discretisation. For details, the reader is referred to [49, 38, 50, 19]. After obtaining the discretised equation set, FV discretisation of dynamic pressure terms will be given in detail. Dynamic pressure terms require special attention due to jump conditions across the free surface, Eqn. (14) and Eqn. (15), which shall be treated using the GFM.

3.2. Momentum equation

The discretised, integral form of the momentum equation, Eqn. (13) reads:

$$\left\{\frac{\partial \mathbf{u}}{\partial t}\right\} + \{\nabla\cdot(\mathbf{u}\mathbf{u})\} - \{\nabla\cdot(\nu_e \nabla \mathbf{u})\} = -\beta \nabla p_d \,, \tag{18}$$

where the time derivative, convection and diffusion term are discretised implicitly. Usual linearisation procedure of the convection term is employed to avoid non–linearity [50]. Special considerations are not needed for implicit terms in Eqn. (18) because they do not have a discontinuity at the interface (see Eqn. (4) and Eqn. (5)), while standard FV discretisation of the pressure gradient term will be given in detail in Sec. 3.5, followed by jump condition corrections presented in Sec. 4.

On assembly, Eqn. (18) is represented by a linear equation for each cell:

$$a_P \mathbf{u}_P + \sum_f a_N \mathbf{u}_N = \mathbf{b} \,, \tag{19}$$

where a_P represents the diagonal coefficient and a_N the off–diagonal coefficient for the equation arising from discretisation for a control volume P, $\sum_f$ denotes sum over all neighbouring faces, subscripts P and N denote field values ($\mathbf{u}$ in this case) defined in cell centres P and N, respectively. $\mathbf{b}$ is the source term arising from old time step contribution in the time derivative term, possible non–orthogonal correction in the diffusion term [38], deferred correction on convection [50] and explicit dynamic pressure gradient term. Over–relaxed correction approach [38, 51] is exclusively used in present work for non–orthogonal correction in diffusion terms. Boundary cells have additional diagonal and/or source contributions arising from boundary conditions [38].

3.3. Pressure equation

The pressure equation is used to create conservative fluxes in incompressible fluid flow. Segregated pressure–velocity coupling is achieved following Patankar and Spalding [42]. Using notation by Jasak [38], the derivation of the pressure equation starts from semi–discretised form of the mixture momentum equation, Eqn. (19):

$$a_P \mathbf{u}_P = \mathbf{H}(\mathbf{u}_N) - \beta \nabla p_d \,. \tag{20}$$

In analogy to the Rhie and Chow correction [52], the dynamic pressure gradient term is left undiscretised to facilitate derivation of the pressure equation. $\mathbf{H}(\mathbf{u}_N)$ term consists of two parts: the transport part containing matrix coefficients for all neighbours multiplied by

corresponding velocities; and the source part $\mathbf{b}$ as in Eqn. (19), but excluding the pressure gradient term:

$$\mathbf{H}(\mathbf{u}_N) = -\sum_f a_N \mathbf{u}_N + \mathbf{b}\,. \tag{21}$$

Discretised, integral form of the continuity equation, Eqn. (7) reads:

$$\int_{CV} \nabla.\mathbf{u}\,\mathrm{d}V = \int_{\partial CV} \mathrm{d}\mathbf{S}.\mathbf{u} = \sum_f \mathbf{s}_f.\mathbf{u}_f = 0\,, \tag{22}$$

where the first identity follows from Gauss' theorem and the second identity follows from second–order accurate polyhedral FV discretisation [38]. Index f denotes face centred value. The velocity field can be defined using the semi–discretised momentum equation, Eqn. (20) as:

$$\mathbf{u}_P = \frac{\mathbf{H}(\mathbf{u}_N)}{a_P} - \frac{\beta \nabla p_d}{a_P}\,. \tag{23}$$

Cell centred velocities given by Eqn. (23) are used to obtain face centred velocities by linear interpolation:

$$\mathbf{u}_f = \frac{(\mathbf{H}(\mathbf{u}_N))_f}{(a_P)_f} - \left(\frac{1}{a_P}\right)_f (\beta)_{f_\Gamma} (\nabla p_d)_{f_\Gamma}\,, \tag{24}$$

where $()_f$ denotes cell–to–face interpolation, and $()_{f_\Gamma}$ denotes cell–to–face interpolation with correction at the interface Γ due to jump conditions, Eqn. (14) and Eqn. (15). Interface–corrected interpolation based on the GFM proposed in this paper will be presented in detail in Sec. 4. Eqn. (24) is later used to calculate face fluxes with a new pressure field p_d that enforces the continuity equation.

The pressure equation is obtained by substituting face centred velocities given by Eqn. (24) into the discretised continuity equation, Eqn. (22):

$$\sum_f \mathbf{s}_f \cdot \left(\frac{1}{a_P}\right)_f (\beta)_{f_\Gamma} (\nabla p_d)_{f_\Gamma} = \sum_f \mathbf{s}_f \cdot \frac{(\mathbf{H}(\mathbf{u}_N))_f}{(a_P)_f}\,. \tag{25}$$

We shall proceed by inspecting individual terms of the pressure equation at the free surface:

- $(1/a_P)_f$ represents face interpolated inverse diagonal coefficient of the momentum equation. The momentum equation is continuous across the free surface (see Sec. 2.2), hence its diagonal does not have a jump. Under such circumstances, ordinary interpolation is sufficient;
- $(\beta)_{f_\Gamma}$ represents face interpolated inverse density. Density field has a discontinuity across the free surface, hence, ordinary linear interpolation is not sufficient;
- $(\nabla p_d)_{f_\Gamma}$ represents the surface normal gradient of the dynamic pressure. Since dynamic pressure has a jump at the free surface, ordinary interpolation schemes are not sufficient;
- $(\mathbf{H}(\mathbf{u}_N))_f$ is smooth across the free surface because it arises from the velocity field in the momentum equation, which is itself continuous.

We stress that even though β and ∇p_d are discontinuous at the interface, their product is continuous according to Eqn. (15). The continuity of $\beta \nabla p_d$ has an important implication: interface corrected interpolation schemes $()_{f_\Gamma}$ obtained via GFM for density and dynamic pressure

have to preserve the symmetry of the pressure Laplacian in Eqn. (25), making the resulting matrix symmetric.

Using Eqn. (24), volumetric face flux is calculated as:

$$F = \mathbf{s}_f \cdot \mathbf{u}_f = \mathbf{s}_f \cdot \left(\frac{(\mathbf{H}(\mathbf{u}_N))_f}{(a_P)_f} - \left(\frac{1}{a_P} \right)_f (\beta)_{f_\Gamma} (\nabla p_d)_{f_\Gamma} \right), \tag{26}$$

after solving the pressure equation, Eqn. (25), making volumetric face fluxes conservative in the discrete form: $\sum_f F = 0$. Conservative volumetric fluxes are used to convect α, $\mathbf{u}$ and other transported variables.

3.4. VOF equation

The discretised form of the VOF transport equation, Eqn. (17) reads:

$$\left\{ \frac{\partial \alpha}{\partial t} \right\} + \{\nabla \cdot (\mathbf{u}\alpha)\} + \{\nabla \cdot (\mathbf{u}_r \alpha(1-\alpha))\} = 0\,, \tag{27}$$

where all terms are treated implicitly. The compressive velocity field $\mathbf{u}_r$, oriented towards the interface in the normal direction, is defined as [53]:

$$\mathbf{u}_r = c_\alpha \hat{\mathbf{n}}_\Gamma \frac{CFL_{ref} |\mathbf{d}_f|}{\Delta t}\,, \tag{28}$$

where c_α is a compression constant used to control the sharpness of the interface (usually taken as one) and $\hat{\mathbf{n}}_\Gamma$ is the unit normal vector to the interface Γ. Reader is referred to Rusche [19] for details regarding the numerical calculation of the unit normal vector from the VOF field. CFL_{ref} is the reference compression CFL number usually taken as 0.5, $|\mathbf{d}_f|$ is the distance between cell centres sharing an internal face, and Δt is the time step size. Note that the sharpness of the interface only affects effective viscosity, Eqn. (4), while the jump in dynamic pressure and density shall be modelled appropriately using the GFM in Sec. 4.

3.5. Finite Volume discretisation of pressure terms

We shall proceed with analysis of dynamic pressure terms. Dynamic pressure p_d and inverse density β are present only in two places, always appearing as a product: the source term in the momentum equation, Eqn. (20) and the pressure Laplacian in the pressure equation, Eqn. (25).

3.5.1. Gauss gradient discretisation

Using Gauss' theorem, second–order accurate discretised pressure gradient for a control volume P reads:

$$\begin{aligned} \beta_P \nabla p_{dP} &= \frac{\beta_P}{V_P} \sum_f \mathbf{s}_f p_{df_\Gamma} \\ &= \frac{\beta_P}{V_P} \sum_f \mathbf{s}_f \left(f_x p_{dP} + (1 - f_x) p_{dN} \right)_\Gamma\,, \end{aligned} \tag{29}$$

where the second identity is obtained by linear interpolation of cell centred values and $f_x = \overline{fN}/\overline{PN}$ is the central–differencing weight [38]. However, index Γ indicates that one–sided extrapolation will be used to obtain p_{dP} and p_{dN} for cells near the interface. Second–order accurate, one–sided extrapolation is obtained from jump conditions Eqn. (14) and Eqn. (15) in Sec. 4 following the GFM approach given by Huang *et al.* [11].

3.5.2. Least squares gradient discretisation

Following Jasak and Weller [54], least squares pressure gradient reads:

$$\beta_P \nabla p_{dP} = \beta_P \sum_f w_f^2 \mathbf{G}^{-1} \cdot \mathbf{d}_f (p_{dN} - p_{dP})_\Gamma \,, \tag{30}$$

where index Γ denotes one–sided extrapolation for p_{dN} and p_{dP} near the free surface, $w_f = 1/|\mathbf{d}_f|$ is the least squares weight and $\mathbf{G}$ is a 3×3 symmetric tensor:

$$\mathbf{G} = \sum_f w_f^2 \mathbf{d}_f \mathbf{d}_f \,. \tag{31}$$

Least squares evaluation of the gradient is second–order accurate irrespective of local arrangement of neighbouring cells [54], making it a favourable choice for unstructured, skewed grids. It is clear that both schemes can be easily adapted to account for pressure jump conditions.

3.5.3. Pressure Laplacian discretisation

Discretisation of the pressure Laplacian on the LHS of Eqn. (25) reads [38]:

$$\sum_f \mathbf{s}_f \cdot \left(\frac{1}{a_P}\right)_f (\beta)_{f_\Gamma} (\nabla p_d)_{f_\Gamma} = \sum_f \left(\frac{1}{a_P}\right)_f (\beta)_{f_\Gamma} |\mathbf{s}_f| \frac{(p_{dN} - p_{dP})_\Gamma}{|\mathbf{d}_f|} + \sum_f \mathbf{k} \cdot \left(\frac{1}{a_P}\right)_f (\beta \nabla p_d)_f^o \,. \tag{32}$$

The first term on the RHS of Eqn. (32) denotes the implicit contribution of the surface normal gradient. The second term denotes its explicit non–orthogonal correction, defined in terms of old, face interpolated pressure gradient $(\beta \nabla p_d)_f^o$. The pressure gradient term pre–multiplied by inverse density is continuous, Eqn. (15) and safe to interpolate from cell centres to face centres. The non–orthogonal correction vector $\mathbf{k}$ is obtained using the over–relaxed approach [38]. We stress that discretisation of the pressure equation given by Eqn. (32) uses a compact computational stencil: control volume P interacts only with its immediate neighbours N.

3.6. Turbulence modelling

Turbulence modelling in CFD is most often achieved with two–equation eddy viscosity turbulence models [47], while Direct Numerical Simulation (DNS) for marine hydrodynamics is still out of reach [55]. Other notable possibilities for turbulence modelling are Large Eddy Simulations (LES) [56] or Detached Eddy Simulations (DES) [57], which should provide more accurate, but increasingly time–consuming solutions. The free surface method presented here allows general turbulence modelling (two–equation, LES and DNS) without algorithmic change. This is straightforward because the effect of turbulence is modelled through two–phase kinematic eddy viscosity used in the momentum equation. Here, we will use $k - \omega$ SST two–equation model [58] which proved to be accurate for steady resistance computations [59]. Although two–equation models are mostly developed for single–phase flows, their transport is mostly influenced by the velocity field through convective term and local production (source) term. As the velocity field is continuous across the interface and the velocity gradient is approximated as continuous, it is suitable to use single–phase turbulence models in standard marine hydrodynamics CFD. Moreover, turbulent kinetic energy k and specific dissipation

of turbulent kinetic energy ω do not directly depend on the pressure and density fields in incompressible flow, which have a discontinuity across the interface.

For details of the $k-\omega\ SST$ model, the reader is referred to Menter [58], and to Menter *et al.* [40] for an overview of the industrial experience with the model.

3.7. Coupling of 6–DOF motion equations with the flow solution

In order to calculate sinkage and trim of a ship at steady state, it is necessary to introduce rigid body motion equations into the solution algorithm. Yang and Lohner [60] used a quasi–static approach that solves for sinkage and trim increment at desired time–step and updates the grid correspondingly. Although their idea is elegant, we use 6–DOF rigid body motion equations as presented in [6] in order to be able to simulate more complex motions in the future.

To facilitate the evaluation of rigid body motion, a moving, ship fixed reference frame is introduced as in [6], making the moment of inertia tensor constant in time and diagonal. The flow field is solved in the global Cartesian, inertial coordinate system, allowing calculation of forces and moments acting on a floating body:

$$\mathbf{F} = \mathbf{F}_p + \mathbf{F}_v\,, \tag{33}$$

$$\mathbf{M} = \mathbf{M}_p + \mathbf{M}_v\,, \tag{34}$$

where index p stands for pressure and index v stands for viscous force and moment. Forces and moments are calculated in the global coordinate system as:

$$\mathbf{F}_p = \sum_{bf} \mathbf{s}_f p_f\,, \tag{35}$$

$$\mathbf{F}_v = \sum_{bf} \rho_f \nu_{e,f} \mathbf{s}_f \cdot \mathbf{T}^*\,, \tag{36}$$

$$\mathbf{M}_p = \sum_{bf} \mathbf{r}_f \times \mathbf{s}_f p_f\,, \tag{37}$$

$$\mathbf{M}_v = \sum_{bf} \mathbf{r}_f \times \left(\rho_f \nu_{e,f} \mathbf{s}_f \cdot \mathbf{T}^*\right)\,, \tag{38}$$

where $\sum_{bf}$ denotes summation over all body (ship) faces, ρ_f is the corresponding density at the boundary face and $\nu_{e,f}$ is the effective viscosity. $\mathbf{T}^*$ represents the deviatoric part of the stress tensor T, which is defined as twice the symmetric part of the $\nabla\mathbf{u}$ tensor. r_f is the distance vector from current boundary face to centre of gravity of the body.

After obtaining forces and moments in the global coordinate system, they are transformed into a body fixed coordinate system using quaternions [61]. Using quaternions compared to Euler angles is more suitable for a general motion as they prevent gimbal lock phenomenon. Time evolution of quaternions involves only 4 additional differential equation for rigid body rotation, as opposed to 9 equations for time evolution of the orthogonal transformation tensor [61]. Hence, a total of 13 differential equation are solved:

- 6 equations for rigid body translation (3 for velocity and 3 for displacement),
- 7 equations for rigid body rotation (3 Euler equations for rotational velocity and 4 for rotation angles using quaternions).

Ordinary differential equations are solved with the adaptive step size Embedded Runge–Kutta 5th order method with Cash–Karp parameters [62]. Compared to the fluid flow solver, the cost of additional 13 ordinary differential equations is negligible.

Solution of ordinary differential equations yields floating body displacement and velocity. Displacement is used to move the computational grid as a rigid body. As the control volumes are now moving in space and time, grid motion flux for each face is calculated using the Space Conservation Law (SCL) [63] and is subtracted from the absolute flux calculated in global coordinate system. Reader is referred to Ferziger and Perić [50] for details on FV discretisation of the SCL using Euler time integration schemes. Apart from grid motion, rigid body velocity is calculated and specified as a boundary condition for all boundary faces on the floating body.

The coupling between rigid body motion and fluid flow is significant as it moves the computational domain and prescribes velocity boundary conditions for the floating body. Simonsen *et al.* [64] report using 5 fluid flow–rigid body motion correctors per time–step for heave and pitch simulations of a ship in regular head waves. However, as the steady state solution is sought in this case, only one corrector is used here.

4. Discretisation of pressure jump conditions using the Ghost Fluid Method

FV discretisation of dynamic pressure terms near the free surface presented in Sec. 3 requires one–sided extrapolates of β and p_d. Before detailed derivation and analysis, the following assumptions are made, where the level set model presented by Huang *et al.* [11] is extended to VOF approach:

- Cell P is considered wet or dry, if $\alpha_P > 0.5$ or $\alpha_P < 0.5$, respectively;

- If a wet cell is completely surrounded by other wet cells, usual discretisation practices (see Sec. 3) are employed, since the free surface is not located in the immediate vicinity of the cell. Similarly, a dry cell completely surrounded with other dry cells does not require special attention.

Such treatment is possible because arbitrary polyhedral FV method uses a compact computational stencil: interface corrections need to be employed for faces where $\alpha_P > 0.5$ and $\alpha_N < 0.5$ or vice versa.[1] Faces that require special treatment are called interface faces because the free surface is located somewhere between P and N cell centres. The following criterion for interface faces holds:

$$(\alpha_P - 0.5)(\alpha_N - 0.5) < 0\,. \tag{39}$$

Eqn. (39) is used to mark all interface faces after solving the VOF transport equation, Eqn. (27). Interface faces are presented in Figure 2 for a uniform 2–D grid for clarity.

[1] If the free surface position is tracked using LS or other approaches, equivalent free surface detection criterion are used.

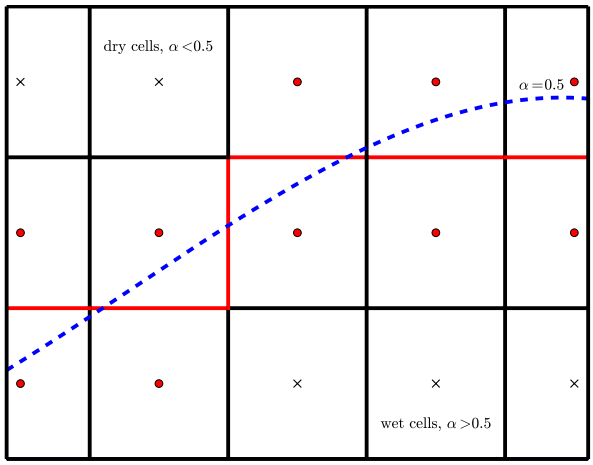


Figure 2: Interface faces for a uniform 2–D grid. The dashed blue line denotes free surface location: $\alpha = 0.5$. Wet cells are below the blue line: $\alpha > 0.5$, while dry cells are above the blue line: $\alpha < 0.5$. Interface faces are represented with red lines. Interface face is a face where the free surface is located between adjacent cell centres. Ordinary faces are denoted with black lines.

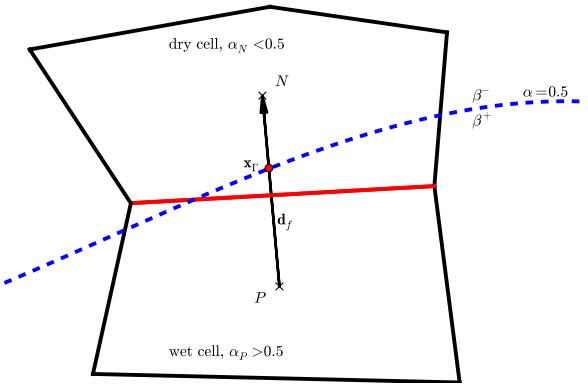


Figure 3: Compact computational stencil near the free surface. Interface face shared by cells P and N is marked with a red line. $\mathbf{x}_\Gamma$ is the location of the free surface for this interface face.

4.1. Computational stencil near the free surface

Before we proceed with the GFM procedure for defining one–sided extrapolates of β and p_d for interface faces using dynamic pressure jump conditions, we shall define necessary geometrical data for computational stencil at the free surface. Figure 3 presents the computational stencil at the free surface in 2–D for clarity, without loss of generalisation. Wet cell P shares a common interface face with dry cell N and d_f is the vector from cell centre P to cell centre N. Huang *et al.* [11] use dimensionless distance to the free surface according to the LS field as:

$$\lambda = \frac{\phi_P}{\phi_P - \phi_N}\,, \tag{40}$$

where ϕ_P and ϕ_N denote signed distance function to the interface at cell centres P and N, respectively. To avoid calculation of the signed distance field ϕ, we define a VOF variant of dimensionless distance:

$$\lambda = \frac{\alpha_P - 0.5}{\alpha_P - \cancel{0.5} - \alpha_N + \cancel{0.5}} = \frac{\alpha_P - 0.5}{\alpha_P - \alpha_N}\,. \tag{41}$$

λ is used to calculate the location of the free surface:

$$\mathbf{x}_\Gamma = \mathbf{x}_P + \lambda \mathbf{d}_f\,. \tag{42}$$

Dynamic pressure jump condition, Eqn. (14) is discretised as:

$$[p_d] = p_d^- - p_d^+ = (\rho^+ - \rho^-)\mathbf{g}\bullet\mathbf{x}_\Gamma = \mathcal{H}\,, \tag{43}$$

where p_d^- is the dynamic pressure at an infinitesimal distance from the free surface on the air side, and p_d^+ its counterpart on the water side. Recall that ρ^+ is water density and ρ^- is air density. The RHS of Eqn. (43) can be calculated and stored as $\mathcal{H}$ for all interface faces at given time–step. The discretisation given by Eqn. (43) is of the same order of accuracy as the solution of VOF transport equation because the dimensionless distance λ

is calculated using the α field.

4.2. One–sided pressure extrapolation from the wet cell

We shall first consider the case where cell P is wet: $\alpha_P > 0.5$. In order to discretise the dynamic pressure gradient jump condition given by Eqn. (15), the procedure by Huang *et al.* [11] is employed:

$$\begin{aligned}[\beta\nabla p_d] &= \beta^-(\nabla p_d)^- - \beta^+(\nabla p_d)^+ \\ &= \beta^- \frac{p_{dN} - p_d^-}{1-\lambda} - \beta^+ \frac{p_d^+ - p_{dP}}{\lambda} = 0\,,\end{aligned} \tag{44}$$

where the first term on RHS denotes a second–order accurate gradient evaluated in the air at the free surface and the second term denotes a second–order accurate gradient evaluated in the water at the free surface. Expressing p_d^- from the discretised dynamic pressure jump condition, Eqn. (43) gives:

$$p_d^- = p_d^+ + \mathcal{H}\,. \tag{45}$$

Inserting Eqn. (45) into Eqn. (44) yields:

$$p_d^+ = \frac{\lambda\beta^-}{\overline{\beta_w}} p_{dN} + \frac{(1-\lambda)\beta^+}{\overline{\beta_w}} p_{dP} - \frac{\lambda\beta^-}{\overline{\beta_w}}\mathcal{H}\,, \tag{46}$$

where:

$$\overline{\beta_w} = \lambda\beta^- + (1-\lambda)\beta^+\,. \tag{47}$$

Here, $\overline{\beta_w}$ represents the weighted inverse density which depends on the distance to the free surface λ; its equation following from algebraic manipulations. Eqn. (46) defines the dynamic pressure at the free surface on the water side p_d^+ in terms of dynamic pressure values at cell centres P and N and the density jump $\mathcal{H}$. Second–order accurate extrapolate of the dynamic pressure at dry cell centre N can be calculated as:

$$p_{dN_\Gamma}^+ = p_d^+ + \frac{1-\lambda}{\lambda}\left(p_d^+ - p_{dP}\right)\,, \tag{48}$$

where $(p_d^+ - p_{dP})/\lambda$ represents dynamic pressure gradient at the free surface on the water side, used to extrapolate dynamic pressure from the free surface towards the dry cell centre N (hence $1-\lambda$). Using Eqn. (46), Eqn. (48) takes the following reduced form after some algebraic manipulations:

$$p_{dN_\Gamma}^+ = \frac{\beta^-}{\overline{\beta_w}} p_{dN} + \left(1 - \frac{\beta^-}{\overline{\beta_w}}\right) p_{dP} - \frac{\beta^-}{\overline{\beta_w}}\mathcal{H}\,. \tag{49}$$

Eqn. (49) defines a second–order accurate extrapolate of the dynamic pressure at the neighbouring cell, obtained with the discretised jump conditions. Hence, the procedure uses jump conditions to define a special interpolation scheme for interface faces. In scientific literature, this method is called the Ghost Fluid Method because $p_{dN_\Gamma}^+$ is defined by one–sided extrapolation of dynamic pressure. $p_{dN_\Gamma}^+$ appears in pressure gradient discretisations, Eqn. (29) and Eqn. (30), and pressure Laplacian discretisation, Eqn. (32).

4.3. One–sided pressure extrapolation from the dry cell

We shall now consider the case where cell P is dry: $\alpha_P < 0.5$. Dynamic pressure jump condition, Eqn. (43) is used to express p_d^+:

$$p_d^+ = p_d^- - \mathcal{H}\,. \tag{50}$$

Discretisation of the dynamic pressure gradient jump condition, Eqn. (15) yields:

$$\beta^{+}\frac{p_{dN}-p_{d}^{+}}{1-\lambda}-\beta^{-}\frac{p_{d}^{-}-p_{dP}}{\lambda}=0\,. \tag{51}$$

Comparing Eqn. (51) with Eqn. (44), it can be seen that β^{+} and β^{-} are interchanged as expected. Inserting Eqn. (50) into Eqn. (51) yields:

$$p_{d}^{-}=\frac{\lambda\beta^{+}}{\overline{\beta_{d}}}p_{dN}+\frac{(1-\lambda)\beta^{-}}{\overline{\beta_{d}}}p_{dP}+\frac{\lambda\beta^{+}}{\overline{\beta_{d}}}\mathcal{H}\,, \tag{52}$$

where:

$$\overline{\beta_{d}}=\lambda\beta^{+}+(1-\lambda)\beta^{-}\,. \tag{53}$$

It should be noted that $\overline{\beta_{d}}\neq\overline{\beta_{w}}$. Following the same procedure as before, second–order accurate extrapolate of the dynamic pressure at wet cell centre N can be calculated as:

$$p_{dN_{\Gamma}}^{-}=p_{d}^{-}+\frac{1-\lambda}{\lambda}\left(p_{d}^{-}-p_{dP}\right)\,. \tag{54}$$

Insertion of Eqn. (52) into Eqn. (54) yields:

$$p_{dN_{\Gamma}}^{-}=\frac{\beta^{+}}{\overline{\beta_{d}}}p_{dN}+\left(1-\frac{\beta^{+}}{\overline{\beta_{w}}}\right)p_{dP}+\frac{\beta^{+}}{\overline{\beta_{w}}}\mathcal{H}\,. \tag{55}$$

Eqn. (55) defines a second–order accurate extrapolate of dynamic pressure if the cell P is dry and consequently, cell N is wet.

4.4. Summary of dynamic pressure extrapolation formulae

In the dynamic pressure extrapolation using jump conditions, we distinguish two cases:

- Cell P is wet and cell N is dry:

$$p_{dN_{\Gamma}}^{+}=\frac{\beta^{-}}{\overline{\beta_{w}}}p_{dN}+\left(1-\frac{\beta^{-}}{\overline{\beta_{w}}}\right)p_{dP}-\frac{\beta^{-}}{\overline{\beta_{w}}}\mathcal{H}\,, \tag{56}$$

$$p_{dP_{\Gamma}}^{-}=\frac{\beta^{+}}{\overline{\beta_{w}}}p_{dP}+\left(1-\frac{\beta^{+}}{\overline{\beta_{w}}}\right)p_{dN}+\frac{\beta^{+}}{\overline{\beta_{w}}}\mathcal{H}\,, \tag{57}$$

- Cell P is dry and cell N is wet:

$$p_{dN_{\Gamma}}^{-}=\frac{\beta^{+}}{\overline{\beta_{d}}}p_{dN}+\left(1-\frac{\beta^{+}}{\overline{\beta_{d}}}\right)p_{dP}+\frac{\beta^{+}}{\overline{\beta_{d}}}\mathcal{H}\,, \tag{58}$$

$$p_{dP_{\Gamma}}^{+}=\frac{\beta^{-}}{\overline{\beta_{d}}}p_{dP}+\left(1-\frac{\beta^{-}}{\overline{\beta_{d}}}\right)p_{dN}-\frac{\beta^{-}}{\overline{\beta_{d}}}\mathcal{H}\,. \tag{59}$$

Eqn. (57) and Eqn. (59) are derived in analogy to Eqn. (56) and Eqn. (58), looking from cell N to cell P. Eqn. (56), (57), (58) and (59) need further comments. In OpenFOAM, cell P is denoted as owner of neighbouring cell N, depending on the direction of the face normal. Consequently, cell P may be wet or dry, leading to expressions for:

- $p_{dN_{\Gamma}}^{+}$ - used when P is wet (N is dry) for discretisation looking from cell P;
- $p_{dP_{\Gamma}}^{-}$ - used when P is wet (N is dry) for discretisation looking from cell N;
- $p_{dN_{\Gamma}}^{-}$ - used when P is dry (N is wet) for discretisation looking from cell P;
- $p_{dP_{\Gamma}}^{+}$ - used when P is dry (N is wet) for discretisation looking from cell N.

Eqn. (56) and Eqn. (57) will be used to prove the symmetry of the pressure Laplacian in Eqn. (32), which is a natural consequence of the incompressibility condition.

4.5. Notes on density extrapolation

Face interpolated density in the pressure equation, Eqn. (32) is extrapolated as follows.

- Looking from the wet cell:

$$(\beta)_{f_\Gamma} = \beta^+ \,, \tag{60}$$

- Looking from the dry cell:

$$(\beta)_{f_\Gamma} = \beta^- \,. \tag{61}$$

Simple expressions for face interpolated densities are also a consequence of the incompressibility condition, where the density field is assumed constant in a given fluid.

4.6. Interface contribution in the discretised Gauss pressure gradient

To demonstrate the effect of interface corrections on the evaluation of the Gauss gradient, we inspect the case where cell P is wet in detail. Eqn. (29) can be written in the following form:

$$\begin{aligned}\beta_P \nabla p_{dP} = &\frac{\beta_P}{V_P} \sum_{f\nparallel} \mathbf{s}_f \left(f_x p_{dP} + (1 - f_x) p_{dN}\right) \\ &+ \frac{\beta_P}{V_P} \sum_{f\Gamma} \mathbf{s}_f \left(f_x p_{dP} + (1 - f_x) p^+_{dN_\Gamma}\right) \,,\end{aligned} \tag{62}$$

where $\sum_{f\nparallel}$ denotes summation over non–interface faces and $\sum_{f\Gamma}$ denotes summation over interface faces. The first sum on the RHS of Eqn. (62) does not need special attention: ordinary interpolation scheme is sufficient. Furthermore, if cell P is wet ($\alpha_P > 0.5$), then $\beta_P = \beta^+$. For interpolation on interface faces (second sum in Eqn. (62)), $p^+_{dN_\Gamma}$ is used because cell P is wet (+). We proceed to examine interface contributions in the second sum. Inserting Eqn. (56) into second sum in Eqn. (62) yields:

$$\begin{aligned}\frac{\beta_P}{V_P} \sum_{f\Gamma} \mathbf{s}_f \left(f_x p_{dP} + (1 - f_x) p^+_{dN_\Gamma}\right) = &\frac{1}{V_P} \sum_{f\Gamma} \mathbf{s}_f \beta^+ p_{dP} \\ &+ \frac{1}{V_P} \sum_{f\Gamma} \mathbf{s}_f \frac{\beta^+ \beta^-}{\overline{\beta_w}} (1 - f_x)(p_{dN} - p_{dP} - \mathcal{H}) \,,\end{aligned} \tag{63}$$

where the first term on the RHS can be identified as first order extrapolation. The second term represents the second order correction arising from discretisation of dynamic pressure jump conditions via GFM. Eqn. (63) represents interface correction contributions for interface faces when the cell is wet. Similar expression can be obtained for dry cell using $p^-_{dN_\Gamma}$ given by Eqn. (58).

4.7. Interface contribution in the discretised least squares pressure gradient

Least squares gradient discretisation for wet cell P, Eqn. (30), divided into two sums reads:

$$\begin{aligned}\beta_P \nabla p_{dP} = &\frac{\beta_P}{V_P} \sum_{f\nparallel} \mathbf{l}_f \left(p_{dN} - p_{dP}\right) \\ &+ \frac{\beta_P}{V_P} \sum_{f\Gamma} \mathbf{l}_f \left(p^+_{dN_\Gamma} - p_{dP}\right) \,,\end{aligned} \tag{64}$$

where $\mathbf{l}_f = w_f^2 \mathbf{G}^{-1} \cdot \mathbf{d}_f$. Inserting Eqn. (56) into the second sum in Eqn. (64) yields:

$$\frac{\beta_P}{V_P} \sum_{f\Gamma} \mathbf{l}_f \left(p^+_{dN_\Gamma} - p_{dP} \right) = \sum_{f\Gamma} \mathbf{l}_f \frac{\beta^+ \beta^-}{\overline{\beta_w}} (p_{dN} - p_{dP} - \mathcal{H}) , \tag{65}$$

where $\beta_P = \beta^+$ has been used because cell P is considered wet (+).

4.8. *Interface contribution in the discretised pressure Laplacian*

Detailed inspection of the pressure equation, Eqn. (25) in Sec. 3.3 led to conclusion that interface corrected interpolation schemes should produce a symmetric matrix. The implicit part of the pressure equation, Eqn. (32) can be divided into two sums:

$$\begin{aligned} \sum_f \left(\frac{1}{a_P} \right)_f (\beta)_{f_\Gamma} |\mathbf{s}_f| \frac{(p_{dN} - p_{dP})_\Gamma}{|\mathbf{d}_f|} = \\ = \sum_{f\nGamma} \left(\frac{1}{a_P} \right)_f (\beta)_f |\mathbf{s}_f| \frac{p_{dN} - p_{dP}}{|\mathbf{d}_f|} \\ + \sum_{f\Gamma} \left(\frac{1}{a_P} \right)_f (\beta)_{f_\Gamma} |\mathbf{s}_f| \frac{p^+_{dN_\Gamma} - p_{dP}}{|\mathbf{d}_f|} , \end{aligned} \tag{66}$$

where wet owner cell P is considered. The first sum on the RHS of Eqn. (66) denotes fully wet or fully dry faces: interface corrections are not needed and the resulting matrix contributions are symmetric as in single–phase flows. Consider a contribution for a single wet owner cell P and dry neighbour cell N pair, by inserting Eqn. (56) into Eqn. (66):

$$\begin{aligned} \left(\frac{1}{a_P} \right)_f (\beta)_{f_\Gamma} |\mathbf{s}_f| \frac{p^+_{dN_\Gamma} - p_{dP}}{|\mathbf{d}_f|} = \\ = \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} (p_{dN} - p_{dP} - \mathcal{H}) , \end{aligned} \tag{67}$$

where $(\beta)_{f_\Gamma} = \beta^+$ because cell P is considered wet. Diagonal contribution in discretisation of pressure Laplacian at an interface face can be identified as:

$$d_P = - \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} . \tag{68}$$

Upper matrix coefficient accounting for the influence of neighbouring cell N to cell P is:

$$a_{PN} = \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} . \tag{69}$$

Furthermore, additional source term is present on the RHS of the pressure equation due to discretisation of jump conditions:

$$S_P = \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} \mathcal{H} . \tag{70}$$

We now proceed to inspect interface face contribution to the pressure equation for the neighbouring cell N:

$$\begin{aligned} \left(\frac{1}{a_P} \right)_f (\beta)_{f_\Gamma} |\mathbf{s}_f| \frac{p^-_{dP_\Gamma} - p_{dN}}{|\mathbf{d}_f|} = \\ = \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} (p_{dP} - p_{dN} + \mathcal{H}) , \end{aligned} \tag{71}$$

where Eqn. (57) for $p^-_{dP_\Gamma}$ is used and $(\beta)_{f_\Gamma} = \beta^-$ because cell N is considered dry. Diagonal contribution for cell N can be identified as:

$$d_N = - \left(\frac{1}{a_P} \right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+ \beta^-}{\overline{\beta_w}} . \tag{72}$$

Lower matrix coefficient accounting for the influence of owner cell P to cell N is:

$$a_{NP} = \left(\frac{1}{a_P}\right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+\beta^-}{\overline{\beta_w}} \,. \tag{73}$$

Comparison of the upper matrix coefficient given by Eqn. (69) and the lower matrix coefficient given by Eqn. (73) proves the symmetry of the matrix, as postulated in Sec. 3.3. Additional source term for cell N is:

$$S_N = -\left(\frac{1}{a_P}\right)_f \frac{|\mathbf{s}_f|}{|\mathbf{d}_f|} \frac{\beta^+\beta^-}{\overline{\beta_w}} \mathcal{H} \,. \tag{74}$$

It is important to note that additional source terms arising from dynamic pressure jump conditions at interface faces are antisymmetric (see Eqn. (70) and Eqn. (74)). The source terms represent additional flux through the interface face that will be balanced by the dynamic pressure jump after solving the pressure equation. Equivalent properties can be easily derived for dry cell P and wet cell N pair.

By inspecting Eqn. (68) and Eqn. (69), we stress that the matrix diagonal can be reconstructed using off–diagonal matrix coefficients as:

$$d_P = -\sum_f a_{PN} \,, \tag{75}$$

where $\sum_f$ is the summation over all neighbours of cell P, both for interface faces and non–interface faces.

5. Segregated solution algorithm

The flow chart of the segregated solution algorithm is presented in Figure 4. The solution algorithm is a combination of SIMPLE [42] and PISO [43] algorithms. The SIMPLE loop handles $\alpha - \mathbf{u} - p_d$ coupling with the rigid body motion, while the embedded PISO loop is used to further resolve $\mathbf{u} - p_d$ coupling. The SIMPLE loop starts by calculating the 6–DOF rigid body motion, updating the mesh and relative fluxes and solving the VOF transport equation, Eqn. (27). After obtaining a new estimate of the α field, interface faces are marked in order to prepare for discretisation of the dynamic pressure terms in mixture equations, see Sec. 4. Effective viscosity of the two phases is updated using Eqn. (4) before the mixture momentum equation. The PISO loop starts with the solution of momentum equation, Eqn. (19), using the dynamic pressure from the previous iteration. The dynamic pressure equation, Eqn. (25) is formulated and solved, taking into account pressure–density coupling through GFM treatment of interface jump conditions. New dynamic pressure with jumps across the interface is used to correct the velocity field, making the face flux field conservative, Eqn. (26). On convergence of the PISO loop, turbulence model equations are solved and the effective viscosity ν_e is updated.

6. Test cases

As the mathematical model is exact for inviscid flow without surface tension, an inviscid 2–D free surface flow over a ramp [65] is considered first, where the water height at the outlet boundary is compared with the analytical solution. Simulations are carried out using

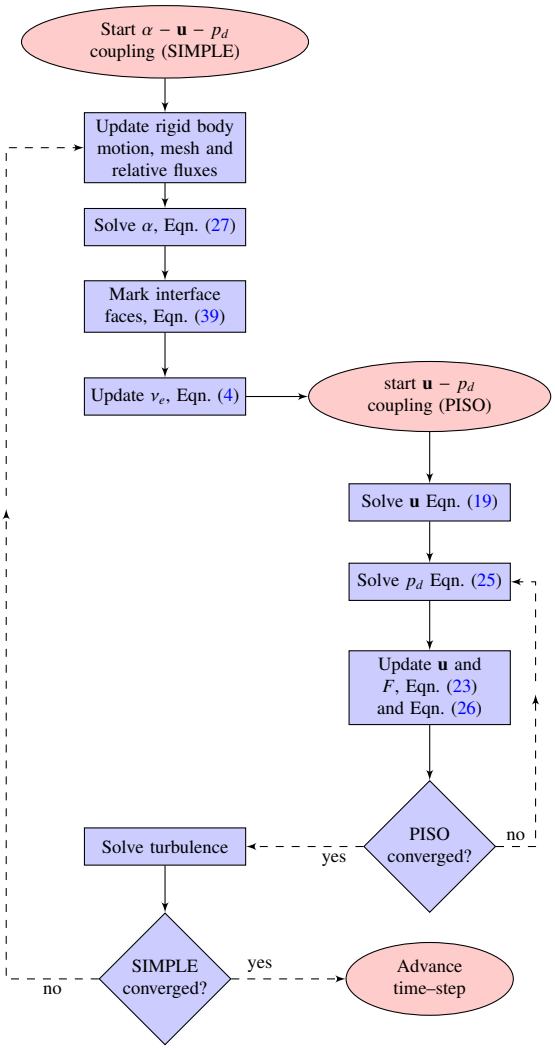


Figure 4: Flow chart of the segregated solution algorithm.

sets of block–structured hexahedral and unstructured prismatic grids in order to compare numerical uncertainties. Furthermore, a simple hydrostatic test case is performed on a block–structured grid in order to discuss spurious velocities which can be observed when using conditionally averaged equations and segregated solution algorithms.

The second set of test cases considers steady resistance of a container ship with dynamic sinkage and trim for different Froude numbers. Four unstructured computational grids are used and detailed verification is performed for all Froude numbers. Results are compared to experimental data published at the Tokyo 2015 Workshop website [66, 67, 68]. Along with iterative and grid uncertainties, validation uncertainties are calculated for all conditions, as the experimental uncertainty is readily available [66].

6.0.1. Discretisation and numerical settings

Time derivative terms are discretised using the first order accurate implicit Euler scheme [50, 49], convective term in the momentum equation is discretised with second order accurate linear upwind scheme, while the convective term in the VOF equation is discretised with second order accurate Total Variation Diminishing (TVD) [69] scheme with van Leer's flux limiter [70] in a deferred correction formulation [50]. Pressure gradient term in the momentum equation is discretised using the second order accurate least squares scheme with interface jump correction relying on the GFM. Pressure Laplacian in the pressure equation is discretised with second order central differencing scheme with interface correction, using the over–relaxed approach for non–orthogonal correction [38, 51]. Since all spatial discretisation schemes are second order accurate, second order grid refinement convergence is expected.

All governing equations are solved with Krylov subspace linear system solvers [71]. Symmetric pressure equation is solved with the Conjugate gradient method [72] using Cholesky factorisation as a preconditioner. Non–symmetric momentum and volume frac-

tion equations are solved with Stabilised Bi–Conjugate Gradient (BiCGStab) [73] method with Incomplete L–U (ILU) preconditioner without fill in [71]. Under–relaxation [42] factors not used unless otherwise stated.

6.0.2. *Outline of validation and verification procedures*

In order to validate the implemented methodology, simulations results shall be compared with analytical and experimental data for first and second test case, respectively. The relative error of the CFD solution S_{CFD} compared to the analytical or experimental solution S is defined as:

$$\epsilon = \frac{S - S_{CFD}}{S}, \tag{76}$$

The order of grid convergence is calculated following guidelines by Versteeg and Malalasekera [49] based on work by Roache [74]:

$$p_G = \frac{\ln\left(\frac{S_f - S_m}{S_m - S_c}\right)}{\ln(r_G)}, \tag{77}$$

where p_G is the achieved order of spatial accuracy, S_c is the coarse grid solution, S_m is the medium grid solution and S_f is the fine grid solution. r_G is the grid refinement ratio, which should preferably be constant. As the constant grid refinement ratio is difficult to achieve using unstructured grids, the average grid refinement ratio between coarse/medium and medium/fine grids is used. We also stress that one of the Roache's [74] conditions for Eqn. (77) to be valid states that the flow field must be sufficiently smooth. This requirement is not met as the dynamic pressure and density have a discontinuity across the free surface. However, since the grid convergence of integral quantities shall be assessed in present work, we assume Eqn. (77) to be valid even though the integral quantities (*i.e.* force on the ship's hull) are obtained using discontinuous density and pressure fields. Following guidelines given by Stern *et al.* [75], Richardson extrapolation (RE) is used to calculate normalised grid uncertainty:

$$\overline{U}_G = \frac{U_G}{|S_f|} = \frac{\delta_{RE}}{|S_f|}, \tag{78}$$

where S_f is the finest grid solution and δ_{RE} is the grid error:

$$\delta_{RE} = \frac{\left|\epsilon_{G_{m,f}}\right|}{r_G^{|p_G|} - 1}, \tag{79}$$

where $\epsilon_{G_{m,f}} = S_f - S_m$ is the difference between fine and medium grid solutions.

It is important to note that grid uncertainty calculated with Eqn. (78) is valid only if monotone convergence is achieved. For oscillatory grid convergence, order of accuracy cannot be calculated. However, following Stern *et al.* [75], grid uncertainty may be estimated as:

$$\overline{U}_G = \frac{U_G}{|S_f|} = \frac{1}{2}\frac{|S_{max} - S_{min}|}{|S_f|}, \tag{80}$$

where S_{max} and S_{min} are the maximum and minimum solutions obtained from a set of coarse, medium and fine grids.

Finally, if neither monotonically converging or oscillatory converging solutions are obtained with grid refinement, grid uncertainty is estimated following Si-

monsen *et al.* [64]:

$$\overline{U}_G = \frac{U_G}{|S_f|} = \frac{|S_{max} - S_{min}|}{|S_f|}, \tag{81}$$

calculating the grid uncertainty as the absolute value of the maximum and minimum solutions.

For the steady resistance test cases, experimental data uncertainty U_D is reported in [66] for all Froude number conditions in percentages of measured values. Hence, it is possible to assess overall validation uncertainty as:

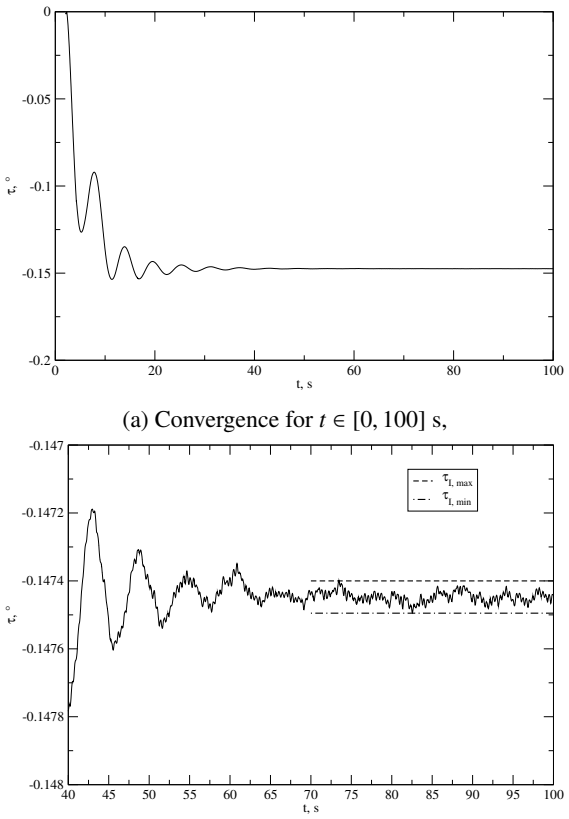
$$U_V = \sqrt{U_D^2 + U_{SN}^2}, \tag{82}$$

where U_{SN} is the simulation numerical uncertainty calculated from iterative uncertainty U_I and grid uncertainty U_G:

$$U_{SN} = \sqrt{U_I^2 + U_G^2}. \tag{83}$$

Convergence of the drag force coefficient, sinkage and trim in time is observed to be oscillatory for the steady resistance simulations. Figure 5 presents the oscillatory convergence of trim for largest Froude number condition. Figure 5a presents trim signal from simulation start ($t = 0$ s) to end ($t = 100$ s), while Figure 5b presents zoomed convergence from $t = 40$ s to end of simulation. Although the convergence is oscillatory, oscillations occur within a narrow band, indicating low iterative uncertainty. In order to quantify iterative uncertainty, we use the expression analogous to Eqn. (80):

$$\overline{U}_I = \frac{U_I}{\epsilon_{f-m}} = \frac{1}{2}\frac{|S_{I,max} - S_{I,min}|}{\epsilon_{f-m}}, \tag{84}$$



(a) Convergence for $t \in [0, 100]$ s,

(b) Convergence for $t \in [40, 100]$ s.

Figure 5: Representative signal of trim convergence for the KCS ship in calm sea at $F_r = 0.282$ (case 6).

where $S_{I,max}$ is the maximum value and $S_{I,min}$ is the minimum value. Both $S_{I,max}$ and $S_{I,min}$ are taken from last few hundred time–steps (iterations) since small oscillations often exhibit irregular behaviour. As an example, Figure 5b presents illustration of $\tau_{I,max}$ and $\tau_{I,min}$ obtained from the time–evolution signal of τ. Iterative uncertainty is always calculated based on the fine grid solution. $\epsilon_{f-m} = S_f - S_m$ denotes the difference between the fine and the medium grid solution and allows U_I to be represented in a dimensionless form.

Table 1: Simulation parameters for the 2–D ramp test case [76].

Item	Value
ρ_h, kg/m^3	1
ρ_l, kg/m^3	0.001
$\mathbf{g}$, m/s^2	$[0, -9.81, 0]$
h_1, m	1
$\mathbf{u}$, m/s	$[6, 0, 0]$
$F_r = \lvert\mathbf{u}\rvert / \sqrt{\lvert\mathbf{g}\rvert h_1}$	1.92

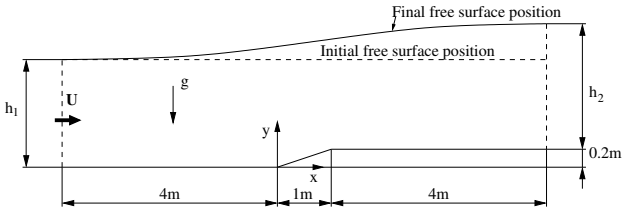


Figure 6: Geometry of the computational domain for the 2–D ramp test case.

Table 2: Boundary conditions for the 2–D ramp test case.

	Inlet	Outlet	Bottom	Top
$\mathbf{u}$	*f.v.*	*z.g.*	*s.*	*s.*
p_d	*z.g.*	*z.g.*	*z.g.*	*f.v.*
α	*f.v.*	*z.g.*	*z.g.*	*z.g.*

6.1. *Inviscid free surface flow over a 2–D ramp*

The inviscid flow model is easily obtained by setting kinematic viscosities of two fluids to zero. Validation for test case presented in this section is achieved by comparing simulation results with analytical solution. Verification is achieved by performing grid refinement studies for both block–structured hexahedral grids and unstructured prismatic grids.

Steady state flow over a 2–D ramp is a standard validation test case for free surface flows used by many authors [65, 76], where the geometry of the computational domain is presented in Figure 6. Applying the Bernoulli and the continuity equation for a known inflow velocity $\mathbf{u}$ and free surface height h_1, it is possible to obtain the height of the free surface at the outlet boundary h_2 [76]. Simulation parameters are outlined in Table 1.

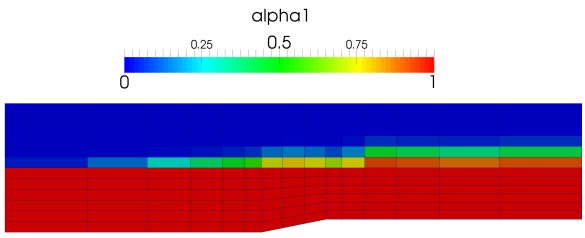
For a general variable ϕ, following key words are used for certain boundary conditions:

- Zero gradient (*z.g.*): $\mathbf{n}.\nabla\phi = 0$,
- Fixed value (*f.v.*): $\phi = \phi_b$,
- Slip (*s.*): $\mathbf{t}.(\nabla\mathbf{u}) = \mathbf{0}$ and $\mathbf{n}.\mathbf{u} = 0$,

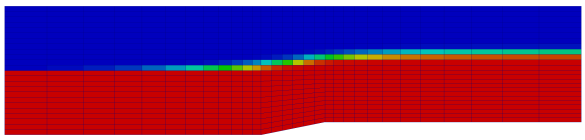
where ϕ_b is the specified value at the boundary and $\mathbf{t}$ denotes unit tangential vector. Boundaries of the computational domain include: inlet, outlet, bottom and top (see Figure 7), with corresponding boundary conditions given in Table 2. As the flow is inviscid, slip boundary conditions are used for velocity at the bottom boundary. At the top, dynamic pressure is set to zero. Uniform velocity field corresponding to the inflow velocity $\mathbf{u}$ and undisturbed free surface with constant height of h_1 are used as initial conditions.

6.1.1. *Refinement study with block–structured hexahedral grids*

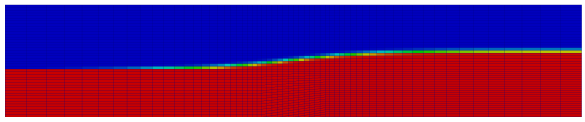
The initial, coarsest grid consists of $15 \times 12 = 180$ cells, shown in Figure 7a. Grading of cells in longitudinal direction towards the ramp and in the vertical direction towards the undisturbed free surface is used. A constant grid refinement ratio [74] $r_G = 2$ is applied three times, producing three additional grids consisting of 720, 2 880 and 11 520 cells. Steady state solution for each of four grids is presented in Figure 7. The



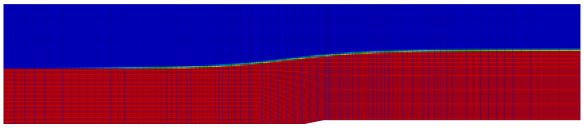
(a) $15 \times 12 = 180$ cells,



(b) $30 \times 24 = 720$ cells,



(c) $60 \times 48 = 2\,880$ cells,



(d) $120 \times 96 = 11\,520$ cells.

Figure 7: Volume fraction α at the steady state solution for four structured hexahedral grids.

computed free surface height at the outlet boundary is compared with the analytical solution $h_{2a} = 1.08973$ m [76] for all grids. Table 3 presents the solution for four grids and corresponding relative errors calculated with Eqn. (76).

Taking into account three coarsest grids (1, 2 and 3), achieved order of spatial accuracy is $p_G = 1.55$, while taking into account three finest grids (2, 3 and 4) results in $p_G = 6.87$. The unrealistically high order of convergence using three finest grids is due to the fact that the solutions are not within asymptotic range of convergence, as calculated using the Grid Convergence Index following guidelines presented in [77]. Taking into account three coarsest grids (1,2 and 3), grid uncertainty is $\overline{U}_G = 0.67\%$, while for the three finest grids $\overline{U}_G \approx 10^{-4}\%$.

Table 3: Structured grid refinement results for 2–D ramp test case.

Index	1	2	3	4
No. cells	180	720	2 880	11 520
h_2, m	1.03114	1.07241	1.08650	1.08662
ϵ, %	5.38	1.59	0.30	0.29

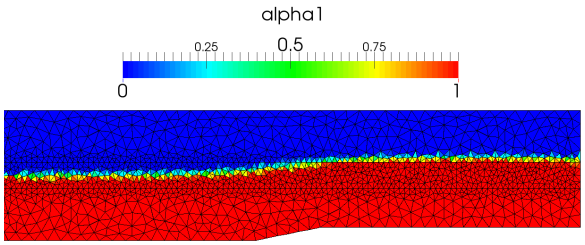
Table 4: Unstructured grid refinement results for 2–D ramp test case.

Index	1	2	3
No. cells	2 892	13 913	26 112
h_2, m	1.02634	1.09777	1.09250
ϵ, %	5.82	-0.73	-0.25

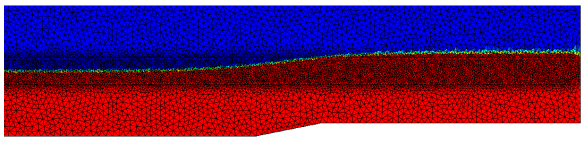
6.1.2. *Refinement study with unstructured prismatic grids*

Three grids with prisms (extruded triangles in the third, dummy direction) are used for the unstructured grid refinement study. The coarse, medium and fine grids consist of 2 892, 13 913 and 26 112 cells, respectively, leading to an average grid refinement ratio $r_G \approx 1.78$. Figure 8 presents the steady–state α field and provides visual details of the grids. Table 3 presents the water height at the outlet for the three grids and corresponding relative errors.

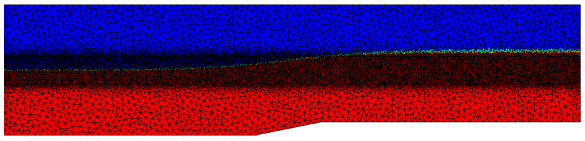
Although the relative errors using unstructured grids reported in Table 4 are similar to the relative errors obtained with structured grids, monotone convergence is not achieved for unstructured grids. The achieved order of spatial accuracy can not be calculated, thus, we proceed to inspect grid uncertainties. Calculating grid uncertainties using Eqn. (80) for oscillatory convergence,



(a) 2 892 cells,



(b) 13 913 cells,



(c) 26 112 cells,

Figure 8: Volume fraction α at the steady state solution for four unstructured prismatic grids.

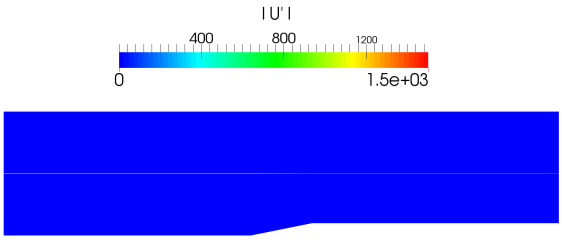
we obtain $\overline{U}_G = 3.3\%$, which is higher than in the case of structured grids.

6.1.3. Spurious air velocities in hydrostatic test case

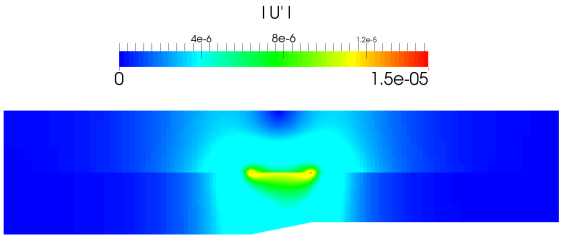
As indicated in Sec. 1.3, the numerical model using conditionally averaged momentum equation with segregated solution algorithms yields spurious air velocities. Specifically, the cause of spurious air velocities can be directly linked to the pressure–density coupling being resolved in the momentum equation. We stress that this numerical phenomena is unrelated to parasitic currents caused by numerical issues in atomisation calculations due to Continuous Surface Stress (CSS) model [46]. This is demonstrated by considering the inviscid case without surface tension effects.

To test the improvements using the present approach, we consider the hydrostatic variant of this test case where the calm free surface is initialised and the velocity field is set to zero everywhere. Such hydrostatic test case is linear, hence the converged solution should be obtained in a single time–step. OpenFOAM's `interFoam` multiphase flow solver is based on conditionally averaged equations and is used for comparison. Additional details regarding the `interFoam` solver can be found in numerous publications [25, 26, 27, 28, 29]. To further stress the inconsistency, we use the fine structured hexahedral mesh.

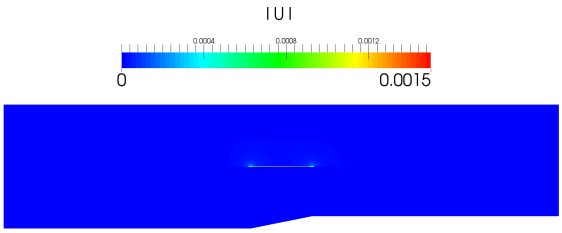
As the considered problem is linear, a single time–step with one pressure correction step has been computed. Velocity fields after the momentum equation (momentum predictor step) and after the pressure correction step are compared in Figure 9 using the two approaches, where we stress that the different velocity scales have been used for each figure in order to visualise the difference. Figure 9a presents the velocity field after the solution of the momentum equation using the conditional averaging approach, where a single layer of cells in air adjacent to the free surface has velocity magnitudes up to $O(10^3)$. The extreme velocities in the intermediate step of the solution algorithm are caused by the dynamic pressure and density (free surface) imbalance in the momentum equation. In the present approach, the dynamic pressure–density (free surface) coupling is resolved in the pressure equation, indicating that there should be no spurious air velocities after the momentum predictor step. This is demonstrated in Figure 9b, where the maximum velocity is



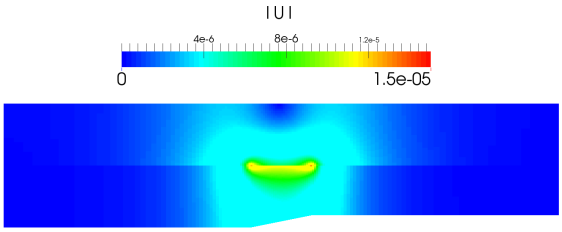
(a) Velocity field after the solution of the momentum equation, conditionally averaged equations.



(b) Velocity field after the solution of the momentum equation, present GFM free surface model.



(c) Final velocity field after pressure correction step, conditionally averaged equations.



(d) Final velocity field after pressure correction step, present GFM free surface model.

Figure 9: Air velocities near the free surface for the hydrostatic test case.

$O(10^{-5})$ due to discretisation errors. The final velocity field after the pressure correction step using conditional averaging approach yields velocity magnitudes of $O(10^{-3})$, which is 7 orders of magnitude smaller than the velocity field obtained after the momentum predictor step. In contrast, the final velocity field using the present approach is the same as after the momentum predictor step, as can be seen by comparing Figure 9d and Figure 9b. Although the maximum velocity magnitude obtained with conditionally averaged approach is reduced by 7 orders of magnitude in the pressure correction step, the final maximum value is two orders of magnitude smaller in the present approach. It should also be noted that the velocity field of $O(10^{-5})$ in the present approach can be observed both in air and water (indicating the level of discretisation errors), while in the conditionally averaged approach, higher velocities can be found only in air cells adjacent to the free surface.

6.1.4. Summary of the results

For the 2–D ramp test case, both block–structured hexahedral and unstructured prismatic grids have been used. For the block–structured hexahedral grids, the achieved order of spatial accuracy for the water height at the outlet is 1.55 considering first three grids and 6.87 considering last three grids (where the solutions are not within asymptotic range of convergence). The results on unstructured grids exhibit oscillatory convergence, hence, the achieved order of spatial accuracy could not be determined. The results on both structured and unstructured grids compare well with the analytical solution. For comparison with the conditionally averaged equations, a simple hydrostatic test case is carried out using the same geometry, where it was demonstrated that the

spurious air velocities (related to density–pressure coupling inside the momentum equation using segregated solution algorithm, not the surface tension effects) are remedied with present approach.

6.2. *Simulations of a Kriso (Moeri) Container Ship in calm seas*

Steady resistance of a ship in calm seas is one of standard test cases for validation and verification of a viscous CFD algorithm, as used by many authors [39, 11, 60, 78]. Viscous free–surface flow around the KCS model free to sink and trim will be considered. KCS is a modern container ship investigated both experimentally [79, 80] and numerically [81], and is one of the three ships used for Tokyo 2015 Workshop on CFD in Ship Hydrodynamics [66]. Experimental data concerning resistance, sinkage and trim, along with the data uncertainty is available at the Workshop's website [66]. Main particulars of the ship for sinkage and trim results are briefly given in Table 5, while the reader is referred to [66] for additional details. Longitudinal centre of buoyancy LCB is $1.48\%L_{PP}$ from mid–ship towards aft of the ship.

Experimental settings are: $g = 9.81$ m/s^2, $\rho_w = 999.5$ kg/m^3 and $\nu_w = 1.27 \cdot 10^{-6}$ m^2/s. The model is free to sink (translation in vertical direction) and trim (rotation around transversal axis), and 6 conditions given in Table 6 are considered, where:

$$F_r = \frac{U}{\sqrt{gL_{PP}}}\,, \tag{85}$$

Table 5: Main particulars of the KCS model [66].

Item	Units	Value
Length between perpendiculars	L_{PP}, m	7.2786
Maximum beam of waterline	B_{WL}, m	1.0190
Depth	D, m	0.6013
Draft	T, m	0.3418
Displacement	∇, m^3	1.6490
Wetted surface coefficient	S_W/L_{PP}^2	0.1803
Block coefficient	C_B	0.6505
Pitch radius of inertia	r_y, m	1.81965

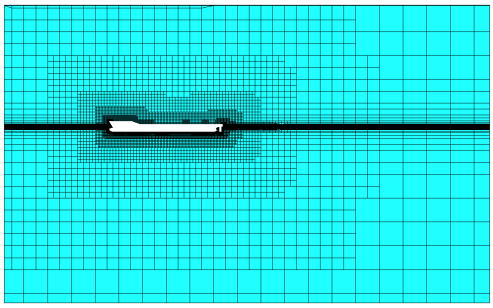
is the Froude number. The Reynolds number is given as:

$$R_e = \frac{UL_{PP}}{\nu_w}\,, \tag{86}$$

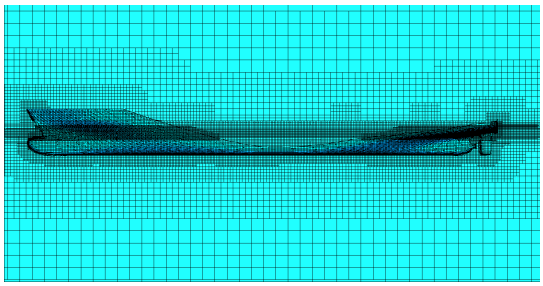
where U is the forward speed of the model.

6.2.1. *Unstructured computational grids*

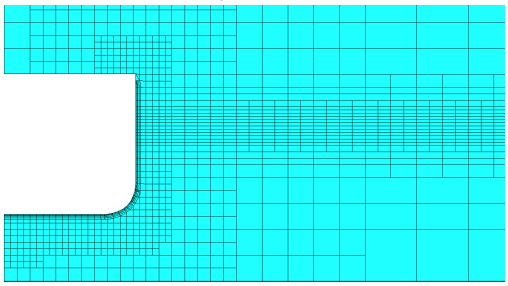
Four unstructured, polyhedral FV grids are used. The origin of the coordinate system is located at mid–ship in the longitudinal direction and at calm water in the vertical direction. Symmetry plane is used to decrease cell count and consequently computational cost. x axis is positive from aft to fore perpendicular of the ship, while y axis is positive towards portside of the ship. z axis is positive upwards, defining a right–handed coordinate system. The bounding box of the computational domain is: $x \in [-21.8, 10.9]$, $y \in [0, 11.4]$ and $z \in [-11.8, 8.2]$ meters. Since grids are unstructured, it is difficult to achieve constant grid refinement ratio. Four grids consist of approximately 600 000, 950 000, 2 600 000 and 4 600 000 cells, and their details are given in Table 7. Grid refinement ratio, $r_G = 1.28$, is calculated as the average of successive grid refinement ratios and is used



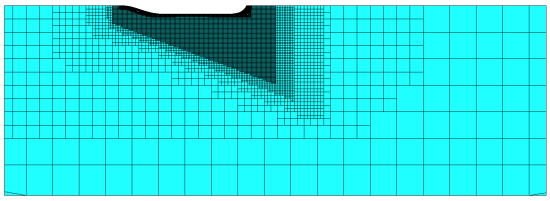
(a) Longitudinal cut (symmetry plane),



(b) Surface grid on the hull,



(c) Transversal cut (midship section),



(d) Vertical cut (still water line),

Figure 10: Coarse KCS grid details.

in the grid convergence study. We note that relatively coarse grids are used with low grid refinement ratios due to limited computational resources. Details of the coarse grid are presented in Figure 10.

6.2.2. *Boundary conditions and numerical settings for KCS cases*

Each grid has the following set of boundaries: inlet, outlet, bottom, atmosphere, far–field, hull and symmetry plane. Corresponding boundary conditions are given in Table 8. $\mathbf{u}_{ship}$ denotes the specified forward speed of the ship, while $\mathbf{u}_{wall}$ denotes moving wall velocity as calculated from the 6–DOF solution. α_{swl} corresponds to the volume fraction distribution according to still water level. Inlet–outlet (mixed) boundary condition is used at the bottom and atmosphere for α: zero gradient is used if outflow occurs, fixed value of 0 and 1 is used otherwise for atmosphere and bottom, respectively. Free stream values of k and ω are calculated and imposed on the inlet, while wall functions are used at the hull. Implicit relaxation zones [82] are used at the inlet, outlet and far–field boundaries.

The following initial conditions are used: uniform velocity field $\mathbf{u} = \mathbf{u}_{ship}$, α distribution corresponding to still water level obtained with cell cutting algorithm, uniform dynamic pressure $p_d = 0$, uniform turbulent kinetic energy $k = k_{fs}$ and uniform specific dissipation of turbulent kinetic energy $\omega = \omega_{fs}$.

Grid motion, k and ω under–relaxation parameters are set to 0.5 and the under–relaxation parameter for velocity field is set to 0.7. Grid motion is under–relaxed explicitly, while $\mathbf{u}$, k and ω are under–relaxed implicitly [42]. α and p_d are not under–relaxed. Under–relaxation parameter of 0.5 for grid motion slightly improves convergence of sinkage and trim compared to under–relaxation parameter of 1 which causes a more

pronounced oscillatory behaviour.

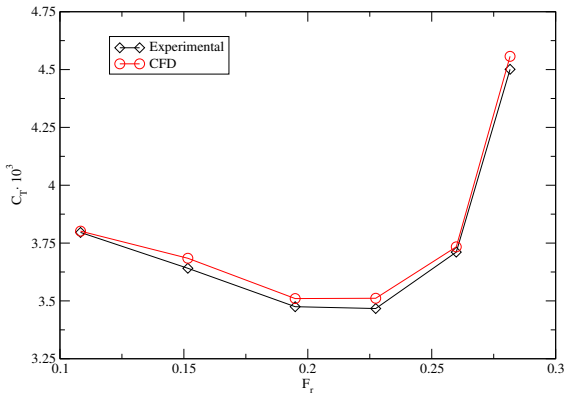
6.2.3. *Computational results for the KCS model in calm seas*

Computations for 6 conditions given in Table 6 are carried out. At steady state, drag force coefficient C_T, sinkage σ and trim τ are calculated and compared to experimental results. Drag force coefficient is calculated as:

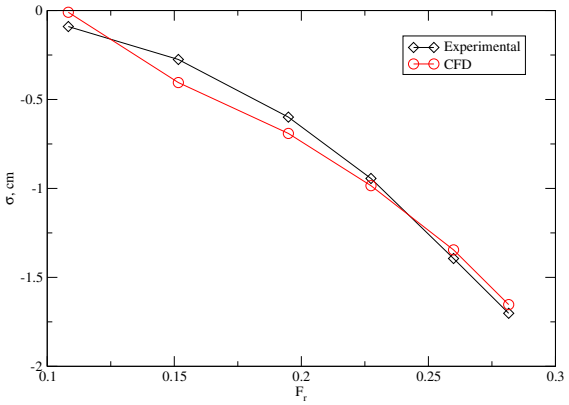
$$C_T = \frac{|F_x|}{0.5\rho_w U^2 S_W}, \tag{87}$$

where F_x is the fluid force acting on a ship hull in the longitudinal direction. Sinkage and trim are readily available as rigid body translation in the vertical direction and rotation around the transversal axis.

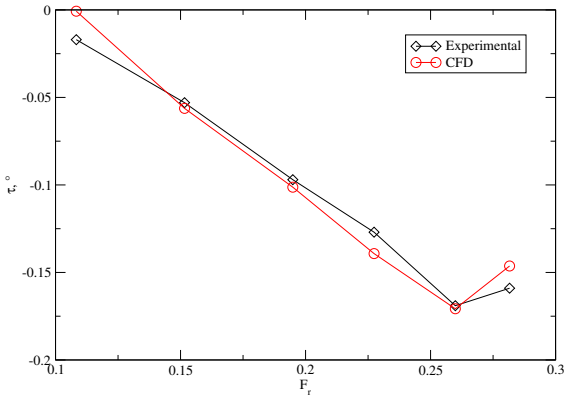
Figure 11 shows the comparison of CFD results on the finest grid (4 600 000 cells) and experimental results over a range of Froude numbers. Figure 11a presents the drag force coefficient, while Figure 11b and Figure 11c show sinkage and trim, respectively. C_T is dimensionless, σ is presented in centimetres for clarity and τ is presented in degrees. CFD results show good agreement compared to experimental results.



(a) Drag force coefficient C_T,



(b) Sinkage σ,



(c) Trim τ.

Figure 11: Comparison of CFD results using the finest grid (4 600 000 cells) and experimental results for 6 KCS test case conditions.

6.2.4. *Verification and validation study*

Verification and validation results corresponding to the grid refinement study for each condition given in Table 6 are presented in Table 9 to 14. CFD results are presented along with corresponding absolute and relative errors compared to experimental results. Iterative and grid uncertainties are calculated for each simulation, and grids 1, 2 and 3 are used for lower Froude number cases: $F_r = 0.108, 0.152, 0.195$, while grids 1, 2, 3 and 4 are used for higher Froude number cases: $F_r = 0.227, 0.260, 0.282$.

Results for the lowest Froude number case, F_r =

Table 6: Steady resistance conditions for the KCS model [66].

Index	i	1	2	3	4	5	6
Forward speed	U, m/s	0.915	1.281	1.647	1.922	2.196	2.379
Froude number	F_r	0.108	0.152	0.195	0.227	0.260	0.282
Reynolds number	$R_e \cdot 10^{-6}$	5.23	7.33	9.42	1.10	1.26	1.36

Table 7: Computational grids for the KCS model.

Grid index	1	2	3	4
Number of cells	607 114	954 016	2 643 763	4 661 889
Hexahedral cells	535 096	853 998	2 422 245	4 331 549
Prismatic cells	23 933	33 430	77 582	118 077
Polyhedral cells	44 599	61 565	132 118	194 026
Maximum non–orthogonality, °	84.3	75.2	86.8	76.1
Average non–orthogonality, °	8.5	7.5	7.5	6.6
Maximum aspect ratio	807	787	656	779

Table 8: Boundary conditions for the KCS test case.

	Inlet	Outlet	Bottom	Atmosphere	Far–field	Hull
$\mathbf{u}$	$\mathbf{u}_{ship}$			pressure outlet	zero gradient	$\mathbf{u}_{wall}$
p_d	zero gradient			fixed value	zero gradient	
α	α_{swl}		inlet–outlet		zero gradient	
k	k_{fs}	zero gradient				wall function
ω	ω_{fs}	zero gradient				wall function

0.108, are presented in Table 9. Although errors for drag force coefficient, sinkage and trim monotonically decrease with grid refinement, convergence is not achieved as the difference between fine and medium grid solutions is larger than the difference between medium and coarse grid solutions. Hence, order of accuracy cannot be calculated. Relative error for the drag force coefficient is less than −5% on the fine grid, while relative errors for sinkage and trim are considerably greater, −110% and −75%, respectively. Larger relative errors for sinkage and trim are expected as the ship exhibits only minor movement at small Froude numbers. Furthermore, experimental measurement of sinkage is slightly less than −0.1 cm, while the finest layer of cells near the free surface is approximately 0.5 cm high. Therefore, we do not expect to correctly model sinkage and trim with such a coarse grid resolution for low Froude numbers. Taking above mentioned considerations into account along with the absolute errors for sinkage and trim of 0.098 cm and 0.013°, respectively, results compare well with experimental data. Iterative uncertainty is low as expected, while the normalised grid uncertainty calculated with Eqn. (81) ranges from 6.4% for sinkage to 16.7% for trim. The normalised

grid uncertainty for the drag force coefficient is 6.8%. Overall validation uncertainties are approximately 0.27, 0.02 and 0.006 for drag force coefficient, sinkage and trim, respectively.

Results for the $F_r = 0.152$ case are presented in Table 10. Errors for all measured parameters monotonically decrease. Relative error for drag force coefficient is less than -5% for the fine grid, while relative errors for sinkage and trim are 40% and -8%. Compared to lowest Froude number case, Table 9, relative errors for sinkage and trim are significantly lowered. This is expected as the same grid better resolves larger motions, and the measured sinkage and trim are approximately 3 times higher compared to $F_r = 0.108$ case. Absolute errors for sinkage and trim are 0.112 cm and 0.004°. Respectively, iterative uncertainties are low: $\approx 10^{-2}$ for the drag force coefficient and $\approx 10^{-5}$ for sinkage and trim. Order of accuracy for the drag force coefficient is 1.85, giving normalised grid uncertainty estimate of 2.7% and validation uncertainty of approximately 0.1.

Results for the $F_r = 0.195$ case are presented in Table 11. Errors for drag force coefficient and sinkage monotonically decrease with grid refinement, while trim exhibits oscillatory convergence. Relative errors for the fine grid: -2% for drag force coefficient, -15% for sinkage and -5% for trim, are further lowered compared to lower Froude number cases. Iterative uncertainties are several orders of magnitude lower than grid uncertainties. Achieved order of accuracy for sinkage is 2.6, giving normalised grid uncertainty estimate of 1.1% and validation uncertainty estimate of 0.03. Normalised grid uncertainty for oscillatory converging trim is 3.8%, followed by validation uncertainty of 0.006. For the drag force coefficient, normalised grid uncertainty is 5.5% with the corresponding validation uncertainty of approximately 0.2.

Results for the $F_r = 0.227$ case are presented in Table 12. Relative errors for the drag force coefficient, sinkage and trim are: -2%, -4.3% and -9.6% on the fine grid, respectively. Drag force coefficient and sinkage exhibit monotone convergence with order of accuracy of 16 and 3.7, respectively, yielding very low grid uncertainty. Iterative uncertainty for the drag force coefficient is -5, due to a minor difference between fine and medium grid solution. Validation uncertainties for the drag force coefficient and sinkage are approximately 0.04 and 0.03. Trim exhibits oscillatory narrow band convergence which gives normalised grid uncertainty estimate of 1.2%. Iterative uncertainty is low as in the previous lower Froude number cases, followed by low validation uncertainty estimate of 0.004.

Results for the design Froude number case, $F_r = 0.260$, are presented in Table 13. Relative errors for the drag force coefficient, sinkage and trim on the fine grid are: -0.4%, 3.5% and -1%, respectively. As in the previous condition, drag force coefficient and sinkage exhibit monotone convergence with 0.86 and 0.13 order of accuracy. Achieved orders of accuracy are lower than the theoretical ones, resulting in higher normalised grid uncertainties: 9.4% for drag force coefficient and 7.6% for sinkage. Combined with low iterative uncertainties, grid uncertainties yield validation uncertainties of 0.35

Table 9: Validation and verification: $F_r = 0.108$ case.

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,°
Experimental results	S	3.796	-0.090	-0.017
Grid 1 solution	S_1	4.242	-0.200	-0.035
Grid 2 solution	S_2	4.141	-0.194	-0.034
Grid 3 solution	S_3	3.972	-0.188	-0.030
Absolute error 1	E_1	-0.446	0.110	0.018
Absolute error 2	E_2	-0.345	0.104	0.017
Absolute error 3	E_3	-0.175	0.098	0.013
Relative error 1	$\epsilon_1, \%$	-11.8	-122.1	-106.1
Relative error 2	$\epsilon_2, \%$	-9.1	-115.6	-97.2
Relative error 3	$\epsilon_3, \%$	-4.6	-109.0	-74.8
Iterative uncertainty	$\overline{U}_I$	$-9.36 \cdot 10^{-3}$	$1.98 \cdot 10^{-5}$	$4.62 \cdot 10^{-6}$
Order of accuracy	p_G	N/A	N/A	N/A
Grid uncertainty	$\overline{U}_G, \%$	6.8	6.4	16.7
Data uncertainty	$U_D, \%$	1.0	8.2	9.6
Validation uncertainty	U_V	0.2730	0.0196	0.0058

Table 10: Validation and verification: $F_r = 0.152$ case.

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,°
Experimental results	S	3.641	-0.275	-0.053
Grid 1 solution	S_1	3.972	-0.423	-0.061
Grid 2 solution	S_2	3.878	-0.414	-0.059
Grid 3 solution	S_3	3.819	-0.387	-0.057
Absolute error 1	E_1	-0.331	0.148	0.008
Absolute error 2	E_2	-0.237	0.139	0.006
Absolute error 3	E_3	-0.178	0.112	0.004
Relative error 1	$\epsilon_1, \%$	-9.1	-53.7	-14.7
Relative error 2	$\epsilon_2, \%$	-6.5	-50.4	-12.0
Relative error 3	$\epsilon_3, \%$	-4.9	-40.4	-7.5
Iterative uncertainty	$\overline{U}_I$	$-5.82 \cdot 10^{-2}$	$1.05 \cdot 10^{-5}$	$1.05 \cdot 10^{-5}$
Order of accuracy	p_G	1.85	N/A	N/A
Grid uncertainty	$\overline{U}_G, \%$	2.7	9.3	7.0
Data uncertainty	$U_D, \%$	1.0	5.9	6.8
Validation uncertainty	U_V	0.1096	0.0426	0.0056

Table 11: Validation and verification: $F_r = 0.195$ case.

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,$^\circ$
Experimental results	S	3.475	-0.599	-0.097
Grid 1 solution	S_1	3.719	-0.709	-0.107
Grid 2 solution	S_2	3.691	-0.697	-0.100
Grid 3 solution	S_3	3.526	-0.691	-0.102
Absolute error 1	E_1	-0.244	0.110	0.010
Absolute error 2	E_2	-0.216	0.098	0.003
Absolute error 3	E_3	-0.051	0.092	0.005
Relative error 1	$\epsilon_1, \%$	-7.0	-18.4	-10.7
Relative error 2	$\epsilon_2, \%$	-6.2	-16.4	-2.8
Relative error 3	$\epsilon_3, \%$	-1.5	-15.3	-5.1
Iterative uncertainty	$\overline{U}_I$	$-5.29 \cdot 10^{-2}$	$3.29 \cdot 10^{-5}$	$-1.69 \cdot 10^{-4}$
Order of accuracy	p_G	N/A	2.57	N/A
Grid uncertainty	$\overline{U}_G, \%$	5.5	1.1	3.8
Data uncertainty	$U_D, \%$	1.0	3.9	4.57
Validation uncertainty	U_V	0.1973	0.0245	0.0060

Table 12: Validation and verification: $F_r = 0.227$ case.

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,$^\circ$
Experimental results	S	3.467	-0.944	-0.127
Grid 1 solution	S_1	3.672	-1.008	-0.146
Grid 2 solution	S_2	3.635	-0.988	-0.138
Grid 3 solution	S_3	3.538	-0.987	-0.141
Grid 4 solution	S_4	3.536	-0.985	-0.139
Absolute error 1	E_1	-0.205	0.064	0.019
Absolute error 2	E_2	-0.168	0.044	0.011
Absolute error 3	E_3	-0.071	0.043	0.014
Absolute error 4	E_4	-0.069	0.041	0.012
Relative error 1	$\epsilon_1, \%$	-5.9	-6.8	-14.6
Relative error 2	$\epsilon_2, \%$	-4.8	-4.7	-8.3
Relative error 3	$\epsilon_3, \%$	-2.1	-4.6	-10.8
Relative error 4	$\epsilon_4, \%$	-2.0	-4.3	-9.6
Iterative uncertainty	$\overline{U}_I$	-5.09	$7.64 \cdot 10^{-4}$	$5.04 \cdot 10^{-4}$
Order of accuracy	p_G	16.0	3.7	N/A
Grid uncertainty	$\overline{U}_G, \%$	0.0	0.2	1.2
Data uncertainty	$U_D, \%$	1.0	2.8	3.3
Validation uncertainty	U_V	0.0359	0.0263	0.0044

(drag force coefficient) and 0.11 (sinkage). Trim exhibits oscillatory convergence with low normalised grid uncertainty of 0.4 and validation uncertainty of 0.004.

Results for the largest Froude number case, F_r = 0.282, are presented in Table 14. Relative errors for drag force coefficient, sinkage and trim on the fine grid are: −1.3%, 2.9% and 8%, respectively. Drag force coefficient and trim exhibit monotone convergence with high orders of accuracy. Achieved order of accuracy for drag force coefficient and trim are approximately 4 and 6, yielding low normalised grid uncertainties of less than 1%. Iterative uncertainties are low as in previous conditions. Resulting validation uncertainties are approximately 0.05 and 0.003. Sinkage exhibits oscillatory convergence within narrow band. Normalised grid uncertainty for sinkage is 0.2%, yielding validation uncertainty of 0.02.

6.2.5. Summary of the results

A set of steady resistance simulations with sinkage and trim is performed for the KCS ship model at different conditions given by a wide range of Froude numbers. Four unstructured grids are used in order to assess grid uncertainty. Iterative uncertainties are also presented for all conditions, and the simulation results are compared to experimental data. Since the uncertainty of experimental data is available, validation uncertainty is assessed as well. Relative errors for the drag force coefficients on fine grids are approximately 5% for two lowest F_r conditions, while the higher F_r conditions yield relative errors less than 2%. Relative errors for sinkage and trim are high for the lowest F_r case as the grid resolution is insufficient to capture minor ship motions. Sinkage and trim relative errors for higher F_r conditions are within 10%.

Grid convergence for the drag force coefficient is obtained for 4 out of 6 conditions: 3 conditions give order of accuracy higher than theoretical, while 1 condition gives order of accuracy lower than theoretical. Normalised grid uncertainty is less than 3%, except for one condition with 10%. Corresponding validation uncertainties are less than 0.5. Monotone grid convergence for sinkage is achieved for 3 out of 6 conditions: 2 conditions yield order of accuracy higher than theoretical, while 1 condition gives order of accuracy lower than theoretical. One condition exhibits oscillatory convergence and others do not converge with grid refinement, even though the errors decrease monotonically. Grid uncertainties for sinkage are less than 1.1%, except for one condition with 7.6%. Corresponding validation uncertainties are approximately 0.03 when the grid convergence is achieved. Trim exhibits oscillatory grid convergence for 3 out of 6 conditions, while 1 condition yields monotone convergence with achieved order of accuracy higher than theoretical. Grid uncertainties are at most 3.8%, yielding largest validation uncertainty of 0.006.

12 out of 18 items (drag force coefficient, sinkage and trim for 6 conditions) exhibit convergence with grid refinement. All 6 non–converging items are clustered at low F_r conditions which are particularly challenging due to low motion magnitudes (0.09 cm for sinkage and 0.017° for trim at lowest Froude numbers) and small

Table 13: Validation and verification: $F_r = 0.260$ case (design condition).

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,°
Experimental results	S	3.711	-1.394	-0.169
Grid 1 solution	S_1	3.981	-1.377	-0.174
Grid 2 solution	S_2	3.910	-1.352	-0.170
Grid 3 solution	S_3	3.808	-1.349	-0.169
Grid 4 solution	S_4	3.726	-1.346	-0.171
Absolute error 1	E_1	-0.270	-0.017	0.005
Absolute error 2	E_2	-0.199	-0.042	0.001
Absolute error 3	E_3	-0.097	-0.045	0.000
Absolute error 4	E_4	-0.015	-0.049	0.002
Relative error 1	$\epsilon_1, \%$	-7.3	1.3	-3.2
Relative error 2	$\epsilon_2, \%$	-5.4	3.0	-0.9
Relative error 3	$\epsilon_3, \%$	-2.6	3.3	-0.3
Relative error 4	$\epsilon_4, \%$	-0.4	3.5	-1.0
Iterative uncertainty	$\overline{U}_I$	$-3.15 \cdot 10^{-1}$	$4.00 \cdot 10^{-3}$	$-7.65 \cdot 10^{-3}$
Order of accuracy	p_G	0.86	0.13	N/A
Grid uncertainty	$\overline{U}_G, \%$	9.4	7.6	0.4
Data uncertainty	$U_D, \%$	1.0	1.9	2.3
Validation uncertainty	U_V	0.3528	0.1056	0.0039

Table 14: Validation and verification: $F_r = 0.282$ case.

Measured parameters		$C_T \cdot 10^3$	σ, cm	τ,°
Experimental results	S	4.501	-1.702	-0.159
Grid 1 solution	S_1	4.837	-1.665	-0.145
Grid 2 solution	S_2	4.737	-1.650	-0.147
Grid 3 solution	S_3	4.608	-1.655	-0.147
Grid 4 solution	S_4	4.560	-1.653	-0.146
Absolute error 1	E_1	-0.336	-0.037	-0.014
Absolute error 2	E_2	-0.236	-0.052	-0.012
Absolute error 3	E_3	-0.107	-0.047	-0.012
Absolute error 4	E_4	-0.059	-0.049	-0.013
Relative error 1	$\epsilon_1, \%$	-7.5	2.2	8.8
Relative error 2	$\epsilon_2, \%$	-5.2	3.1	7.3
Relative error 3	$\epsilon_3, \%$	-2.4	2.8	7.9
Relative error 4	$\epsilon_4, \%$	-1.3	2.9	8.0
Iterative uncertainty	$\overline{U}_I$	$-1.79 \cdot 10^{-1}$	$2.59 \cdot 10^{-4}$	$2.57 \cdot 10^{-3}$
Order of accuracy	p_G	3.98	N/A	6.21
Grid uncertainty	$\overline{U}_G, \%$	0.6	0.2	0.04
Data uncertainty	$U_D, \%$	1.0	1.4	1.8
Validation uncertainty	U_V	0.0541	0.0233	0.0028

wave elevations relative to grid resolution. We believe that better results could be obtained by using finer grids for low F_r conditions, although this is not demonstrated due to limited computational resources. Moreover, it is interesting to note that the oscillatory grid convergence is achieved for 5 out of 12 converging items. For the ramp test case with block–structured grids using constant refinement ratio, monotone convergence has been achieved. Observations regarding oscillatory grid convergence and grid divergence on unstructured grids are in accordance with Larsson *et al.* [15] where better grid refinement results are obtained using structured grids. This is expected as the well controlled, uniform refinement is impossible with unstructured grids. However, low relative errors indicate that accurate results may be obtained with unstructured grids which are appealing for the industrial use due to easier grid generation.

7. Conclusion and future work

An implementation of the Ghost Fluid Method for free surface incompressible flows in the arbitrary polyhedral Finite Volume framework is presented in this paper. The numerical model is presented in detail, with the emphasis on the interface–corrected interpolation schemes for discontinuous pressure and density fields. Interface–corrected schemes are used only for interface faces, while ordinary interpolation is sufficient for fully wet or dry cells. Such treatment is straightforward because polyhedral FV method implemented in OpenFOAM uses compact computational stencil. Discretised Gauss gradient and least squares gradient operators for dynamic pressure are derived and presented. Implicit dynamic pressure Laplacian operator is inspected in detail, proving the symmetry of pressure equation. Furthermore, it follows that the diagonal part of the pressure equation can be reconstructed using negative off–diagonal coefficients as is the case in single–phase flow.

The verification and validation of the model is performed for two sets of test cases: inviscid flow over a 2–D ramp and steady resistance simulations of a container ship for different Froude numbers. The results compare well with analytical and experimental data, while the numerical uncertainties are within reasonably low bounds. The detailed verification and validation exercise renders the presented method suitable for marine hydrodynamic flows.

Incorporation of the surface tension effects in the dynamic pressure jump and the implementation of the viscous jump conditions [14] should enable efficient modelling of the small scale effects, which shall be the topic of future work.

Acknowledgements

This research was sponsored by Bureau Veritas and Hyundai Heavy Industries under the administration of Dr. Šime Malenica and Dr. Geon-Hong Kim, whose support is gratefully acknowledged. The authors express their gratitude towards Johan Rønby and Holger Marschall for numerous helpful discussions and comments. Our gratitude also goes to Mr. Marko Horvat,

a graduate student who provided the grids for the 2–D ramp test case.

References

[1] Z. Tuković, H. Jasak, A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow, Comput. Fluids 55 (2012) 70–84.

[2] K. Dieter-Kissling, H. Marschall, D. Bothe, Numerical method for coupled interfacial surfactant transport on dynamic surface meshes of general topology, Comput. Fluids 109 (2015) 168–184. doi:10.1016/j.compfluid.2014.12.017.

[3] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A Front-Tracking Method for Computations of Multiphase Flow, J. Comput. Phys. 169 (2001) 708–759. doi:10.1006/jcph.2001.6726.

[4] F. Stern, J. Yang, Z. Wang, H. Sadat-Hosseini, M. Mousaviraad, B. S., T. Xing, Computational Ship Hydrodynamics: Nowadays and Way Forward, in: Proceedings of the 29^{th} ONR Symposium on Naval Hydrodynamics, 2012, pp. 1–73.

[5] R. Luppes, B. Düz, H. van der Heiden, P. van der Plas, A. Veldman, Numerical simulations of two–phase flow with COMFLOW: past and recent developments, in: Proceedings of the ECCOMAS 2012 Conference, 2012, pp. 1–16.

[6] P. M. Carrica, R. V. Wilson, R. W. Noack, F. Stern, Ship motions using single–phase level set with dynamic overset grids, Comput. Fluids 36 (2007) 1415–1433. doi:10.1016/j.compfluid.2007.01.007.

[7] P. M. Carrica, R. V. Wilson, F. Stern, An unsteady single–phase level set method for viscous free surface flows, Int. J. Numer. Meth. Fluids 53 (2007) 229–256. doi:10.1002/fld.1279.

[8] R. P. Fedkiw, T. Aslam, S. Xu, The ghost fluid method for deflagration and detonation discontinuities, J. Comput. Phys. 154 (2) (1999) 393–427.

[9] M. Kang, R. P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (3) (2000) 323–360.

[10] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416.

[11] J. Huang, P. M. Carrica, F. Stern, Coupled ghost fluid/two–phase level set method for curvilinear body–fitted grids, Int. J. Numer. Meth. Fluids 44 (2007) 867–897. doi:10.1002/fld.1499.

[12] W. Bo, J. W. Grove, A volume of fluid method based ghost fluid method for compressible multi-fluid flows, Comput. Fluids 90 (2014) 113–122.

[13] M. Kaneda, T. Haruna, K. Suga, Ghost-fluid-based boundary treatment in lattice boltzmann method and its extension to advancing boundary, Appl. Therm. Eng. 72 (1) (2014) 126–134.

[14] B. Lalanne, L. R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, J. Comput. Phys. 301 (2015) 289–307.

[15] L. Larsson, F. Stern, M. Visonneau, Numerical Ship Hydrodynamics: An assessment of the Gothenburg 2010 workshop, Springer, 2013. doi:10.1007/978-94-007-7189-5.

[16] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas-Liquid Multiphase Flows, Cambridge University Press, 2011.

[17] O. Ubbink, R. I. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, J. Comput. Phys. 153 (1999) 26–50.

[18] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. thesis, Imperial College of Science, Technology & Medicine, London (1997).

[19] H. Rusche, Computational fluid dynamics of dispersed two - phase flows at high phase fractions, Ph.D. thesis, Imperial College of Science, Technology & Medicine, London (2002).

[20] K. Kissling, J. Springer, H. Jasak, S. Schütz, K. Urban, M. Piesche, A Coupled Pressure Based Solution Algorithm Based on the Volume–of–Fluid Approach for Two or more Immiscible Fluids, in: V European Conference on Computational Fluid Dynamics, ECCOMAS CFD, 2010.

[21] J. A. Sethian, Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science, Cambridge University Press, 1996.

[22] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Com-

put. Phys. 114 (1994) 146–159.

[23] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246.

[24] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow ii, J. Comput. Phys. 225 (1) (2007) 785–807.

[25] N. G. Jacobsen, D. R. Fuhrman, J. Fredsøe, A wave generation toolbox for the open-source CFD library: OpenFoam®, Int. J. Numer. Meth. Fluids 70 (9) (2012) 1073–1088. doi:10.1002/fld.2726.

[26] P. Higuera, J. Lara, I. J. Losada, Realistic wave generation and active wave absorption for Navier-Stokes models: Application to OpenFoam®, Coast. Eng. 71 (2013) 102–118. doi:10.1016/j.coastaleng.2012.07.002.

[27] P. Higuera, J. Lara, I. J. Losada, Simulating coastal engineering processes with OpenFoam ®, Coast. Eng. 71 (2013) 119–134. doi:10.1016/j.coastaleng.2012.06.002.

[28] B. T. Paulsen, H. Bredmose, H. B. Bingham, An efficient domain decomposition strategy for wave loads on surface piercing circular cylinders, Coast. Eng. 86 (2014) 57–76. doi:10.1016/j.coastaleng.2014.01.006.

[29] B. T. Paulsen, H. Bredmose, H. B. Bingham, N. G. Jacobsen, Forcing of a bottom-mounted circular cylinder by steep regular water waves at finite depth, J. Fluid Mech. 755 (2014) 1–3. doi:10.1017/jfm.2014.386.

[30] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.

[31] M. Sussman, E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (4) (1999) 1165–1191.

[32] P. Gómez, J. Hernández, J. López, On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows, Int. J. Numer. Methods Eng. 63 (2005) 1478–1512.

[33] Y. Sun, C. Beckermann, Sharp interface tracking using the phase–field equation, J. Comput. Phys. 220 (2007) 626–653. doi:10.1016/j.jcp.2007.05.025.

[34] Y. Sun, C. Beckermann, A two–phase diffusive–interface model for Hele–Shaw flows with large property contrasts, Physica D 237 (2008) 3089–3098. doi:10.1016/j.physd.2008.06.010.

[35] S. Wang, J. Glimm, R. Samulyak, X. Jiao, C. Diao, An Embedded Boundary Method for Two Phase Incompressible Flow, ArXiv e-prints.arXiv:1304.5514.

[36] H. Johansen, P. Colella, A Cartesian Grid Embedding Boundary Method for Poisson's Equation on Irregular Domains, J. Comput. Phys. 147 (1998) 60–85. doi:10.1006/jcph.1998.5965.

[37] R. K. Crockett, P. Colella, D. T. Graves, A Cartesian Grid Embedded Boundary Method for Solving the Poisson and Heat Equations with Discontinuous Coefficients in Three Dimensions, J. Comput. Phys. 230 (2010) 613–628. doi:10.1016/j.jcp.2010.12.017.

[38] H. Jasak, Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College of Science, Technology & Medicine, London (1996).

[39] P. Queutey, M. Visonneau, An interface capturing method for free–surface hydrodynamic flows, Comput. Fluids 36 (2007) 1481–1510. doi:10.1002/j.compfluid.2006.11.007.

[40] F. R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, Turb. Heat Mass Tran. 4 (2003) 625–632.

[41] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object oriented techniques, Comput. Phys. 12 (1998) 620–631.

[42] S. V. Patankar, D. B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (1972) 1787–1806.

[43] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1986) 40–65.

[44] F. R. Batchelor, An Introduction to Fluid Dynamics, Cambridge University Press, 1967.

[45] C. Dopazo, On conditional averages for intermittent turbulent flows, J. Fluid Mech. 81 (1977) 433–438.

[46] N. Ashgriz, Handbook of atomization and sprays: theory and applications, Springer Science & Business Media, 2011.

[47] D. C. Wilcox, Turbulence Modeling for CFD, DCW Industries, 1993.

[48] K. Ito, Z. Li, Interface Conditions for Stokes Equations with a

Discontinuous Viscosity and Surface Sources, Appl. Math. Lett. 19 (2006) 229–234. doi:10.1016/j.aml.2005.02.041.

[49] H. K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, Pearson Education Limited, 1995.

[50] J. H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, 1996.

[51] I. Demirdžić, On the Discretization of the Diffusion Term in Finite–Volume Continuum Mechanics, Numer. Heat Transfer, Part B 68 (2015) 1–10. doi:10.1080/10407790.2014.985992.

[52] C. M. Rhie, W. L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.

[53] V. Vukčević, A. Östman, H. Jasak, Rapid Simulations of Pure Sway Motion Using FVM in OpenFOAM, in: SIMMAN 2014: Workshop on Verification and Validation of Ship Manoeuvring Simulation Methods, 2014.

[54] H. Jasak, H. G. Weller, Application of the finite volume method and unstructured meshes to linear elasticity, Int. J. Numer. Methods Eng. 48 (2000) 267–287.

[55] S. C. on CFD in Marine Hydrodynamics, Final report and Recommendations to the 27^{th} ITTC, [Available online; accessed 20 August 2015] (August–September 2014).

[56] F. S. Pereira, G. Vaz, L. Eca, On the Numerical Requirements of RANS and Hybrid Turbulence Models, in: Proceedings of the MARINE 2015 Conference, 2015, pp. 886–902.

[57] D. Li, J. Hallander, T. Johansson, R. Karlsson, Cavitation Dynamics and Underwater Radiated Noise Signature of a Ship with a Cavitating Propeller, in: Proceedings of the MARINE 2015 Conference, 2015, pp. 401–412.

[58] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (8) (1994) 1598–1605.

[59] H. Jasak, V. Vukčević, D. Christ, Rapid Free Surface Simulation for Steady-State Hull Resistance with FVM using OpenFOAM, in: Proceedings of the 30th Symposium on Naval Hydrodynamics, 2014, pp. 548–554.

[60] C. Yang, R. Löhner, Calculation of Ship Sinkage and Trim Using a Finite Element Method and Unstructured Grids, Int. J. CFD 16(3) (2002) 217–227.

[61] E. A. Coutsias, L. Romero, The quaternions with an application to rigid body dynamics, Department of Mathematics and Statistics.

[62] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C++: The Art of Scientific Computing, Cambridge University Press, 2002.

[63] I. Demiridžić, M. Perić, Space conservation law in finite volume calculations of fluid flow, Int. J. Numer. Meth. Fluids 8 (1988) 1037–1050. doi:{10.1002/fld.1650080906}.

[64] C. D. Simonsen, J. F. Otzen, S. Joncquey, F. Stern, EFD and CFD for KCS heaving and pitching in regular head waves, J. Mar. Sci. Technol. 18 (2013) 435–459. doi:10.1007/s00773-013-0219-0.

[65] S. Muzaferija, M. Perić, Computation of free–surface flows using the finite–volume method and moving grids, Numer. Heat Transfer, Part B 32 (1997) 369–384.

[66] Tokyo Workshop, Tokyo 2015: A Workshop on CFD in Ship Hydrodynamics, http://www.t2015.nmri.go.jp/, [Online; accessed 20 August 2015] (2015).

[67] L. Larsson, F. Stern, M. Visonneau, N. Hirata, T. Hino, J. Kim (Eds.), Tokyo 2015: A Workshop on CFD in Ship Hydrodynamics, Vol. 2, NMRI (National Maritime Research Institute), Tokyo, Japan, 2015.

[68] L. Larsson, F. Stern, M. Visonneau, N. Hirata, T. Hino, J. Kim (Eds.), Tokyo 2015: A Workshop on CFD in Ship Hydrodynamics, Vol. 3, NMRI (National Maritime Research Institute), Tokyo, Japan, 2015.

[69] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. 2: Computational Methods for Inviscid and Viscous Flows, John Wiley & Sons, 1984.

[70] B. van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.

[71] Y. Saad, Iterative Methods for Sparse Linear Systems (2nd Edition), Society for Industrial and Applied Mathematics Philadelphia, SIAM, 2003.

[72] M. Hestens, E. Steifel, Method of conjugate gradients for solving linear systems, J. Res. 29 (1952) 409–436.

[73] H. van der Vorst, Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems, SIAM J. Sci. Comput. 13 (1992) 631–644.

[74] P. Roache, Quantification of uncertainty in computational fluid dynamics, Ann. Rev. Fluid. Mech. 29 (1997) 123–160.

[75] F. Stern, R. V. Wilson, H. W. Coleman, E. G. Paterson, Comprehensive Approach to Verification and Validation of CFD Simulations–Part 1: Methodology and Procedures, J. Fluids. Eng 123(4) (2001) 793–802. doi:10.1115/1.1412235.

[76] Z. Tuković, Finite volume method on domains of varying shape (in Croatian), Ph.D. thesis, Faculty of Mechanical Engineering and Naval Architecture, Zagreb (2005).

[77] NPARC Alliance CFD Verification and Validation Web Site, Examining Spatial (Grid) Convergence, http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html, [Online; accessed 1 April 2016] (2016).

[78] C. H. Raven, A. van der Ploeg, B. Starke, Computation of free–surface viscous flows at model and full scale by a steady iterative approach, in: Proceedings of the 25^{th} Symposium on Naval Hydrodynamics, 2004.

[79] S. H. Van, W. J. Kim, G. T. Yim, K. D. H., C. J. Lee, Experimental Investigation of the Flow Characteristics Around Practical Hull Forms, in: Proceedings of the 3rd Osaka Colloguium on Advanced CFD Applications to Ship Flow and Hull Form Design, 1998.

[80] W. J. Kim, S. H. Van, D. Kim, Measurment of flows around modern commercial ship models, Exp. Fluids 31 (2001) 567–578.

[81] C. D. Simonsen, J. F. Otzen, F. Stern, EFD and CFD for KCS heaving and pitching in regular head waves, in: Proceedings of the 27^{th} Symposium on Naval Hydrodynamics, 2008.

[82] H. Jasak, V. Vukčević, I. Gatin, Numerical Simulation of Wave Loads on Static Offshore Structures, in: CFD for Wind and Tidal Offshore Turbines, Springer Tracts in Mechanical Engineering, 2015, pp. 95–105.