A three-dimensional Cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies

Zhihua Xie[a,*], Thorsten Stoesser[b]

[a] *School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK*
[b] *Department of Civil, Environmental and Geomatic Engineering, University College London, UK*

Abstract

A three-dimensional Cartesian cut-cell method for the large-eddy simulation of two-phase flows with moving bodies is presented in this study, which combines a volume-of-fluid method to capture the air-water interface and a moving body algorithm on a stationary, non-uniform, staggered, Cartesian grid. The filtered Navier–Stokes equations are discretised using the finite volume method with the PISO algorithm for velocity-pressure coupling and the dynamic Smagorinsky subgrid-scale model is employed to compute the effect of the unresolved (subgrid) scales of turbulence on the large scales. In the present study, the small cut-cells are unmodified and due to the use of an implicit time integration no instabilities occur during the computations. The versatility and robustness of the present two-phase flow model is illustrated via various two- and three-dimensional flow problems with fixed/moving bodies, such as dambreak flows with and without a square cylinder, a moving cylinder in a quiescent fluid, dambreak flow over a wet bed with a moving gate, water entry and exist of a circular cylinder, and landside-generated

[*] Corresponding author. Email: zhihua.xie@hotmail.com; Tel.: +44(0)29-20879375.

waves. Good agreement is obtained between the numerical results and the corresponding experimental measurements.

Keywords: Cartesian cut-cell method, Finite volume method, two-phase flow, moving bodies, volume-of-fluid method

1. Introduction

Two-phase flows with moving bodies appear in many scientific and engineering applications, e.g. marine renewable energy, hydrodynamics of open channels, naval architecture, hydraulic structures, coastal and offshore engineering, biomedical engineering, oil-and-gas transportation systems, and geophysical flows. These applications typically have the motion of a deformable gas-liquid/liquid-liquid interface and its interaction with a fixed/moving structure. Development of computational methods for predicting such flows, which involve turbulence, breaking waves, air entrainment, impact and fluid-structure interaction is highly challenging.

A key requirement for simulating numerically free-surface/two-phase flows is the tracking or capturing of the interface [1, 2]. Numerous methods have been proposed and used to simulate free-surface/two-phase flows on a fixed mesh, such as marker-and-cell [3], volume-of-fluid (VOF) [1, 4, 5], front-tracking [6], level set [7, 8], phase field [9]. Alternatively, moving mesh [10] and meshless (particle) [11] methods have been proposed. In addition, recent developments include three-phase flow methods [12, 13]. Among these methods, the VOF method, a transport equation for the volume fraction of the two phases, is probably the most popular method on a fixed grid and is widely used due to its inherent properties of: mass conservation, computational efficiency and easy implementation. From a general point of view, there are two classes of algorithms to solve the transport equation of the volume fraction: geometric and algebraic computation [5]. In the geometric VOF methods [1], interfaces are first reconstructed from the volume fraction data so that a geometric profile is found which approximates the actual interface location. Then changes in volume fraction are calculated by integrating volume fluxes across cell boundaries, using flux splitting or unsplitting schemes. In the algebraic computation [14–16], the interface is captured by solving the transport equation of the volume fraction with a differencing scheme without reconstructing the interface, such as the flux-corrected transport scheme [14] and using the normalised variable diagram (NVD) [17] concept to switch between different differencing schemes [15].

Most two-phase flows in engineering applications are turbulent and therefore need different treatment for the turbulence. In many engineering practices, only the time-averaged flow is of interest. Therefore the Reynolds-averaged Navier–Stokes (RANS) equations are usually solved, in which all of the unsteadiness is averaged out and all of the effects turbulence on the mean flow is modelled by one of a number of available turbulence models. As a result, RANS models cannot provide instantaneous flow characteristics. The increase in computer power has led to the development of more powerful but more computationally demanding methods. The most accurate and most straightforward approach is the mthod of direct numerical simulation (DNS), in which the Navier–Stokes equations are solved directly without any modification. All scales of the turbulent flow are captured with DNS, which implies that the grid size must be no larger than the Kolmogorov scale [18].

Due to the high demand of DNS in terms of the number of grid points (which is proportional to $Re^{9/4}$ in 3D, where Re is the Reynolds number), it has been mostly used for relatively low Reynolds number flows or flows in a relatively small flow domain. As DNS is out of reach for practical applications, recent developments have focused on the method of large-eddy simulations (LES) [19, 20], in which the large-scale eddies are resolved, i.e. calculated directly, while the effects of the small eddies on the large-scale turbulence are approximated. LES has been already employed for practical problems where the Reynolds number is high and the computational domain is large.

To deal with complex geometries with fixed or moving motion in engineering applications, overlapping grids, boundary-fitted grids and unstructured grids can be used. Unstructured grids provide great flexibility in conforming to complex boundaries, and can easily be refined or coarsened in specific regions of the flow domain depending on the flow feature. However, they require additional computational efforts and further complicate the algorithm implementation as there is no pre-defined order of the control volumes and their geometric layouts need to be calculated. Furthermore, generating high-quality boundary-fitted or unstructured grids is usually very cumbersome [21], especially for moving body problems, where the mesh has to be regenerated at every time step. Cartesian grid methods which can simulate flow with complex geometries on fixed Cartesian grids, avoid these problems. The primary advantage of the Cartesian grid method is that only little modification of the flow solver on Cartesian grids is needed to account for complex (immersed) geometries. It also has the advantage of simplified grid generation and simulating flows with moving boundaries whilst avoiding deforming grids. The two most popular methods are the immersed boundary method [21–25] and the Cartesian cut-cell method [26–31]. There are some examples for two-phase flows with moving bodies using immersed boundary method in two-dimensional (2D) [32] and three-dimensional (3D) [33] simulations. Compared to immersed boundary methods, the Cartesian cut-cell method is very attractive as it enforces strict conservation of mass, momentum and energy, and in particular near solid boundaries. The present study focuses on the Cartesian cut-cell method hence only relevant studies using this method are mentioned hereafter.

The Cartesian cut-cell method is based on a stationary Cartesian background grid, in which the solid boundary is intersected with boundary cells (named as cut-cells), and regular grid cells are truncated to conform to the solid (immersed) boundary interface. There is no modification for standard regular grid cells whereas special treatments are needed for the cut-cells. For fixed, solid boundaries, the Cartesian cut-cell method has been developed for 2D Poisson [34], and advection-diffusion [35] equations. It has also been applied to study single-phase fluid flow problems, such as for 2D aeroacoustics [36], 2D hypersonic boundary layer transition [37], 2D shallow water equations [38], 2D incompressible viscous flow [27, 28, 31], 2D compressible viscous flow [39], 3D inviscid flow [40], 3D incompressible viscous flow [30], 3D compressible viscous flow [41, 42], and 3D LES studies [43–45]. The Cartesian cut-cell method has also been used to study free-surface/two-phase flows in 2D using a two-fluid approach [46], height function [47], volume-of-fluid method [48–50], in the form of a coupled level-set and volume-of-fluid approach [51], and 3D DNS and LES studies [52, 53].

For moving bodies, Cartesian cut-cell methods have been developed for solving the 2D heat [54] and 2D shallow water equations [55], 2D viscous flow [26], 3D rarefied gas flows [56], 3D inviscid flow [57], and 3D compressible viscous flow [58–60]. Some early development of the cut-cell method for single-phase flows can be found in [29]. For 2D two-phase flows, different approaches have been employed for the interface tracking/capturing, such as the Langrangian marker [61], level-set method [62], and the two-fluid approach [63, 64]. In 3D, the Euler equations have been solved together with a density function to capture the air-water interface for a water impact problem in [65] and a cut-cell method with moving body has been developed to study the 3D wave impact problem with a single-phase volume-of-fluid method in [66]. Recently, there are some development for fluid-structure interaction problems for 2D two-phase flow [67] and 3D single-phase flow [68]. To date, there has been relatively little work on cut-cell methods with moving bodies for the 3D Navier–Stokes equations with two-phase flows and turbulence.

The objective of this paper is, therefore, to present and validate thoroughly an efficient finite volume method based on the Cartesian cut-cell method for the unsteady, turbulent, incompressible, two-phase Navier–Stokes equations with moving bodies on a three-dimensional, non-uniform, staggered, Cartesian grid. The method of large-eddy simulation is employed to compute directly large-scale turbulence of the flow by solving the filtered Navier–Stokes equations and employing the dynamic Smagorinsky sub-grid scale model to account for the unresolved (subgrid) scales of turbulence. An algebraic VOF scheme is employed and modified in cut-cells to capture the air-water interface in the two-phase flow model. Solid boundaries are resolved by the Cartesian cut-cell method, with detailed 3D cut-cell generation and finite volume discretisation. The moving body algorithm (as an internal source function by taking the solid volume change into account) is implemented to simulate moving bodies on a fixed Cartesian grid. Moreover, an implicit time integration scheme is used for time integration, which prevents common instability problems in small cut-cells as reported in the literature.

The organisation of this paper is as follows. The description of the mathematical model for the two-phase flow is described in Section 2. The numerical method and implementation of the cut-cell method are presented in Section 3. The versatility, robustness and accuracy of the present two-phase flow model is demonstrated by solving various 2D and 3D two-phase flow problems with moving bodies in Section 4. Finally, the paper ends with conclusions in Section 5.

2. Mathematical model

2.1. Governing equations

The governing equations used for the incompressible immiscible Newtonian two-phase flow are based on the Navier–Stokes equations, given as:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot [\mu(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla}^T \boldsymbol{u})] + \rho \boldsymbol{g}, \tag{2}$$

where $\boldsymbol{u}$ is the velocity vector, t is the time, p is the pressure, $\boldsymbol{g}$ is the gravitational acceleration vector, ρ and μ are the density and dynamic viscosity of the fluid.

2.2. *Interface modelling*

2.2.1. *Volume-of-fluid method*

The volume-of-fluid method is employed here to capture the air-water interface in the two-phase flow solver during the simulation. F is the volume fraction defined as:

$$F = \begin{cases} 1, & \text{if only water is present;} \\ 0, & \text{if only air is present.} \end{cases} \tag{3}$$

The air-water interface is then within the cells where $0 < F < 1$. A particle on the surface stays on the surface and the volume fraction F has a zero material derivative:

$$\frac{\mathrm{d}F}{\mathrm{d}t} = \frac{\partial F}{\partial t} + \boldsymbol{u} \boldsymbol{\cdot} \boldsymbol{\nabla} F = 0. \tag{4}$$

2.2.2. *Physical properties*

After interface capturing for the volume fraction field, the momentum equation (Eq. (2)) is closed with the constitutive relations for the density and dynamic viscosity of the fluid as given by:

$$\rho = F\rho^{\mathrm{w}} + (1 - F)\rho^{\mathrm{a}}, \tag{5}$$

$$\mu = F\mu^{\mathrm{w}} + (1 - F)\mu^{\mathrm{a}}, \tag{6}$$

where the superscripts ‘w’ and ‘a’ denote water and air, respectively.

2.3. Moving body algorithm

In order the simulate the moving body in a fixed Cartesian grid system, rather than applying a interpolated body force over the grid as used in the IBM, the source function approach [69, 70] is employed here. The motion of a solid can be followed and interacts with the background Cartesian grid as shown in Fig. 1. At time t, considering a cell containing a solid with its volume as $V_{\text{solid}}(t)$, if the volume of solid increases $\partial V_{\text{solid}}(t)/\partial t > 0$, the volume of fluids decreases, and vice versa. Thus, the conservation of mass for the continuity equation (Eq. (1)) in a cell with volume V can be modified as:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = \frac{1}{V}\frac{\partial V_{\text{solid}}(t)}{\partial t} = \psi(t), \tag{7}$$

where $\psi(t)$ is a internal source function depending on the volume change of the solid in the cell at time t. Thus, the momentum equation (Eq. (2)) is also modified as:

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot [\mu(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla}^T \boldsymbol{u})] + \rho \boldsymbol{g} + \rho \boldsymbol{u} \psi(t). \tag{8}$$

2.4. Subgrid-scale model

The large-eddy simulation (LES) approach is adopted in this study, for which the large-scale eddies are solved and a subgrid-scale model is employed to compute the unresolved scales of turbulence. The governing equations used for incompressible two-phase flow are based on the spatially filtered Navier–Stokes equations of Eq. (7)and Eq. (8), given as:

$$\boldsymbol{\nabla} \cdot \bar{\boldsymbol{u}} = \psi(t), \tag{9}$$

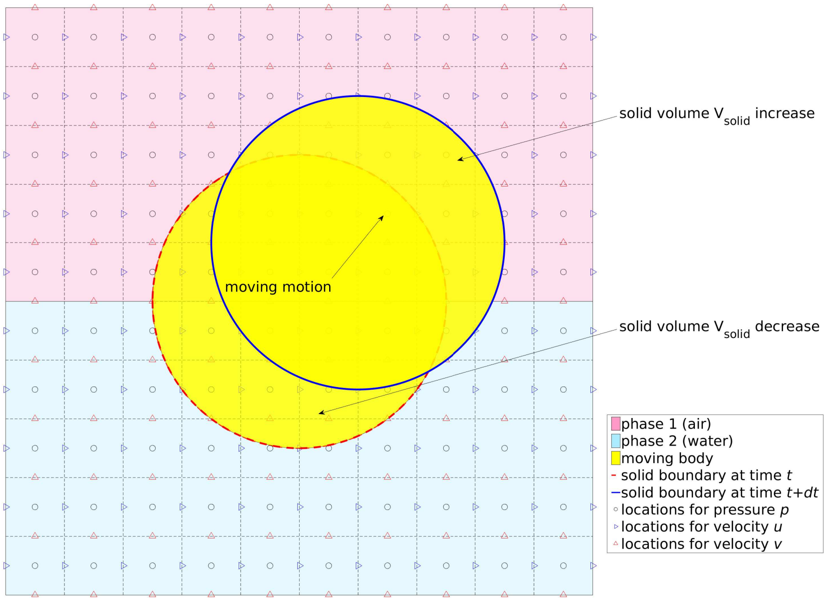

solid volume V$_{solid}$ increase
solid volume V$_{solid}$ decrease
moving motion
phase 1 (air)
phase 2 (water)
moving body
solid boundary at time *t*
solid boundary at time *t*+*dt*
locations for pressure *p*
locations for velocity *u*
locations for velocity *v*

Figure 1: Schematic of 2D cut-cells undergoing a volume change due to boundary motion in a two-phase flow model. The light blue and red colours show the fluid regions for the water and air where the solid domain is represented as yellow region. The yellow area between the red dashed line and blue solid line on the top are cut-cells whose volume of solid increases $\partial V_{\text{solid}}(t)/\partial t > 0$; the yellow area with mesh inside between the red dashed line and blue solid line on the bottom are cut-cells whose volume of solid decreases $\partial V_{\text{solid}}(t)/\partial t < 0$. The arrangement of variables (p, u, v) on a staggered Cartesian grid are also shown, where the velocities are stored on the face of the control volume and the pressure is stored at the centre of the control volume.

$$\frac{\partial(\rho\bar{\boldsymbol{u}})}{\partial t}+\boldsymbol{\nabla}\boldsymbol{\cdot}(\rho\bar{\boldsymbol{u}}\otimes\bar{\boldsymbol{u}})=-\boldsymbol{\nabla}\bar{p}+\boldsymbol{\nabla}\boldsymbol{\cdot}[\mu(\boldsymbol{\nabla}\bar{\boldsymbol{u}}+\boldsymbol{\nabla}^T\bar{\boldsymbol{u}})]+\rho\boldsymbol{g}+\rho\bar{\boldsymbol{u}}\psi(t)+\boldsymbol{\nabla}\boldsymbol{\tau}^{\text{sgs}}, \quad (10)$$

where the overbar $\bar{\cdot}$ denotes the spatial filtering over the grid in Cartesian coordinates (x, y, z), $\bar{\boldsymbol{u}} = (\bar{u}, \bar{v}, \bar{w})$ is the filtered velocity vector and $\bar{p}$ is the filtered pressure.

The term $\boldsymbol{\tau}^{\text{sgs}} = \rho(\bar{\boldsymbol{u}}\bar{\boldsymbol{u}} - \overline{\boldsymbol{u}\boldsymbol{u}})$ is the subgrid-scale (SGS) stress tensor and the anisotropic part of the SGS term is approximated by an eddy-viscosity model of the form [71]:

$$\boldsymbol{\tau}^{\text{sgs}} - \frac{1}{3}\text{trace}(\boldsymbol{\tau}^{\text{sgs}})\boldsymbol{I} = 2\mu_t\bar{\boldsymbol{S}}, \quad (11)$$

where $\boldsymbol{I}$ is the unit tensor and $\bar{\boldsymbol{S}}$ is the strain rate tensor of the resolved field. μ_t is the turbulent eddy viscosity defined as:

$$\mu_t = \rho C_d\bar{\Delta}^2\left|\bar{\boldsymbol{S}}\right|, \text{and} \left|\bar{\boldsymbol{S}}\right| = \sqrt{2\bar{\boldsymbol{S}}\bar{\boldsymbol{S}}}, \quad (12)$$

with the cut-off length scale $\bar{\Delta} = (\Delta x\Delta y\Delta z)^{1/3}$ and the coefficient $C_d = \frac{1}{2}\frac{L_{ij}M_{ij}}{M_{ij}M_{ij}}$ is calculated by the dynamic Smagorinsky model [72] in the present study, where $L_{ij} = \hat{\bar{u}}_i\hat{\bar{u}}_j - \widehat{\bar{u}_i\bar{u}_j}$ and $M_{ij} = \hat{\bar{\Delta}}^2\left|\hat{\bar{S}}\right|\hat{\bar{S}}_{ij} - \bar{\Delta}^2\widehat{\left|\bar{S}\right|\bar{S}_{ij}}$, and the hat $\hat{\cdot}$ represents spatial filtering over the test filter. The symbol for spatial filtering '$^-$' is dropped hereinafter for simplicity. The advantage of the dynamic Smagorinsky model is that the dissipation of energy from the large-scale turbulence is approximated in analogy to dissipation on a molecular level without the need for empirical input (due to the dynamic approximation of the Smagorinsky constant) and hence has found to work well for many applications of practical interest [19].

2.5. Initial and boundary conditions

In order to completely describe the mathematical model it is necessary to define the boundary conditions in a computational domain. For the moving body, the motion is known and the velocity and position of the solid body's boundaries can be prescribed for every time step. For the outlet, the zero-gradient or radiation boundary condition is applied for the flow. As both fluids in the air and water are solved simultaneously in the present two-phase flow model, the kinematic and dynamic free surface boundary conditions are inherently implemented in the VOF method and they do not need to be specified explicitly at the air-water interface.

In the computation, the initial flow field at $t = 0$ has to be prescribed. For calculations with the fluids initially at rest, the flow field is initialised with zero velocity and hydrostatic pressure, and the volume fraction of the air-water interface is computed from the initial water depth. For the moving body, the velocity in the solid is initialised with its moving velocity.

3. Numerical method

3.1. Finite volume discretisation

One option of discretising the governing equations is the finite volume method (FVM). In the FVM, also known as the control volume method, the entire flow domain is divided into a number of control volumes surrounding each grid point. The differential equation is integrated over each control volume (CV) in order to derive the algebraic equation containing the grid-point values of φ, where φ is the dependent variable. The FVM is conservative and can deal with complex geometries [18, 73], thus it is particularly suitable for modelling interfacial flows with moving bodies due to the requirement of mass conservation, the deformed interface and moving boundaries, and therefore it is adopted in the present study.

Consider a volume of fluid Ω which has an arbitrary domain, the surface of the control volume is S and the unit outward normal vector to the face f is $\boldsymbol{n}$. All the governing equations can be recast into a general integral formulation as below

$$\iiint_{\Omega} \frac{\partial}{\partial t}(\rho\varphi)\mathrm{d}\Omega + \iint_{S} (\rho\boldsymbol{u}\cdot\boldsymbol{n})\varphi\mathrm{d}S = \iint_{S} \Gamma\frac{\partial\varphi}{\partial n}\mathrm{d}S + \iiint_{\Omega} Q_{\varphi}^{\mathrm{S}}\mathrm{d}\Omega, \quad (13)$$

where φ denotes the dependent variable, Γ is the viscosity and Q_{φ}^{S} is the source term in the control volume.

Table 1 shows the various values of φ, Γ and Q_{φ}^{S} in the general integral formulation to represent the Navier–Stokes equations and the volume fraction equation. It is noted that the final form of the continuity equation (9) used here is obtained under the assumption that the fluid is incompressible.

Table 1: Values of φ, Γ and Q_{φ}^{S} in the general integral formulation to represent the Navier–Stokes equations.

Equation	φ	Γ	Q_{φ}^{S}
Continuity	1	0	$\rho\psi(t)$
Momentum	$\boldsymbol{u}$	$\mu+\mu_t$	$-\boldsymbol{\nabla}p+\rho\boldsymbol{g}+\rho\boldsymbol{u}\psi(t)$
Volume fraction	F	0	$\rho F\boldsymbol{\nabla}\cdot\boldsymbol{u}$

3.2. Computational grid

In this study, a staggered, Cartesian grid, which has the advantage of strong coupling between the velocity and the pressure is used to discretise the flow domain. Figure 2(a) shows a typical variable arrangement in a 3D Cartesian grid, in which the velocities are located at the centre of the CV's face, and the pressure, all other scalar variables and the volume fraction F are stored at the CV's centre. Figure 2(b) shows a typical control volume used in the present study, in which P is the present grid point (or node), the upper-case letter E, W, N, S, B, and R denote neighbouring nodes on the east, west, north, south, back, and front with respect to the central node P. The lower-case e, w, n, s, b, and r denote the corresponding face of the CV whereas c denotes the centre of the CV.

Thus, the CV's volume is obtained as

$$\Omega = \iiint_{\Omega} \mathrm{d}\Omega = \Delta x \Delta y \Delta z. \tag{14}$$

The area of the face A is similarly calculated, *e.g.,* the one of the east face A_{e} is

$$A_{\mathrm{e}} = \iint_{\mathrm{e}} \mathrm{d}S = \Delta y \Delta z. \tag{15}$$

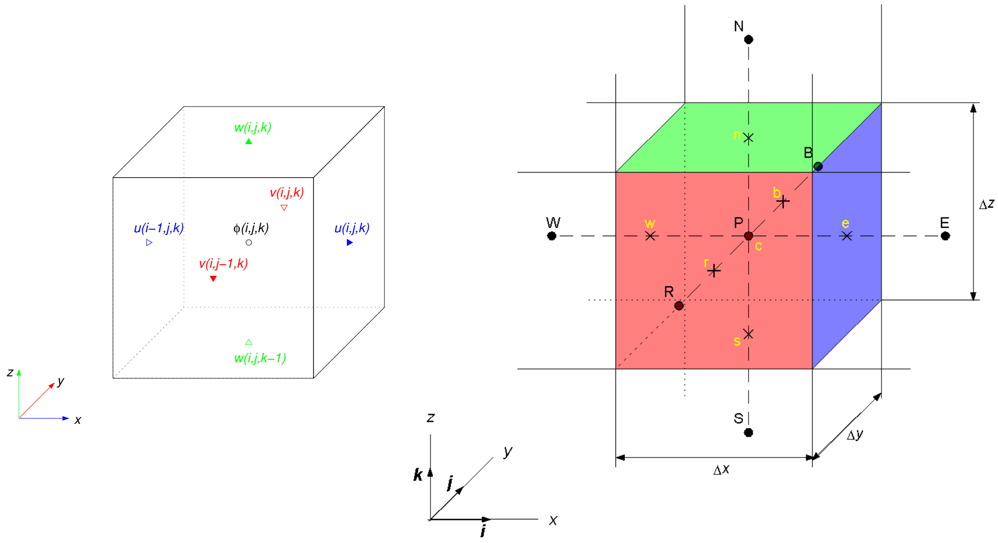
Unless stated otherwise the variable on the face is predicted with linear interpolation

$$\varphi_{\mathrm{e}} = \lambda_{\mathrm{e}} \varphi_{\mathrm{P}} + (1 - \lambda_{\mathrm{e}}) \varphi_{\mathrm{E}}, \tag{16}$$

where λ_{e} is the interpolation factor defined as

$$\lambda_{\mathrm{e}} = \frac{|\mathrm{eE}|}{|\mathrm{PE}|}. \tag{17}$$

Analogous expressions can be derived for all other faces (f=w, n, s, b, r) by making appropriate index substitutions and will not be shown here.



(a) variable arrangement

(b) notation in a control volume

Figure 2: Variables used for the CV (i, j, k) in a 3D staggered grid. Velocities $u(i, j, k)$, $v(i, j, k)$, and $w(i, j, k)$ are stored at the centre of the east, back, and north face of the CV. Pressure and other scalar quantities $\varphi(i, j, k)$ are stored at the centre of the CV. Δx, Δy, and Δz are the CV's length, and $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are the unit vectors in the x, y, and z directions, respectively.

3.3. Navier–Stokes solver

3.3.1. Temporal discretisation

A backward finite difference method is used for the time derivative, which leads to an implicit scheme for the Navier–Stokes equations

$$Q_\varphi^{\mathrm{T}} = \iiint_\Omega \frac{\partial}{\partial t}(\rho\varphi)\mathrm{d}\Omega = \frac{\rho_{\mathrm{c}}^{n+1}\varphi^{n+1} - \rho_{\mathrm{c}}^{n}\varphi^{n}}{\Delta t}\Omega, \tag{18}$$

where Δt is the time step and the superscripts $n + 1$ and n mean the value in current and previous time step, respectively. The implicit scheme has the advantage of unconditional stability and thus can prevent the instability problem in small cut-cells discussed in Section 3.5.

3.3.2. Spatial discretisation

The Navier–Stokes equations are discretised in space using a finite volume formulation on a non-uniform, staggered, Cartesian grid as shown in Figure 2(b). The high-resolution scheme [74] is used for the advection terms. The second-order central difference scheme is used for diffusion terms, pressure gradient terms and the pressure correction equations. However in cut cells, all these disretised terms will need to be modified and this will be discussed in detail in Section 3.5.

Substituting all the discretised terms into Eq. (13) and subtracting the continuity equation $\partial\rho/\partial t + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}) = \rho\psi(t)$ multiplied by $\varphi_{\mathrm{P}}^{n+1}$, leads to

$$a_{\mathrm{P}}^{\varphi}\varphi_{\mathrm{P}}^{n+1} = \sum a_{\mathrm{nb}}^{\varphi}\varphi_{\mathrm{nb}}^{n+1} + b_{\mathrm{P}}^{\varphi}, \tag{19}$$

where a^{φ} is the coefficient, the subscripts P and $\mathrm{nb} = \mathrm{E}, \mathrm{W}, \mathrm{N}, \mathrm{S}, \mathrm{B}, \mathrm{R}$ denote the variables in the present and neighbouring cells respectively and b_{P}^{φ} is the source term.

The algebraic equations are solved by the SIP (Strongly Implicit Procedure) method or Bi-CGSTAB (Bi-Conjugate Gradients Stablized) Method [75] in this study.

3.3.3. Pressure-velocity coupling

In the incompressible Navier–Stokes equations pressure and velocity are decoupled as the pressure term does not appear in the continuity equation. For some numerical discretisations this may cause convergence problems. However, when a staggered mesh is used, as in this work, coupling occurs as a result of the discretisation, as velocity updates on cell faces contain pressure terms. In this study, the PISO algorithm [76] is employed for the pressure-velocity coupling. The PISO algorithm is used to calculate the corrected pressure twice and after solving the pressure correction equations, the updated pressure and velocity are added by the pressure and velocity correction terms respectively. Here only a brief summary is presented and more details can be found in [77]

For a guessed pressure distribution p^*, the discretised momentum equations can be solved to produce the fluid velocities $\boldsymbol{u}^*$, which satisfy

$$a_{\mathrm{P}}^{\boldsymbol{u}} \boldsymbol{u}_{\mathrm{f}}^* = \sum a_{\mathrm{nb}}^{\boldsymbol{u}} \boldsymbol{u}_{\mathrm{nb}}^* + b_{\mathrm{P}}^{\boldsymbol{u}} + A_{\mathrm{f}}(p_{\mathrm{P}}^* - p_{\mathrm{nb}}^*). \tag{20}$$

To obtain the pressure correction, the updated fluid velocities are substituted into the discretised continuity equation Eq. (9) and the resulting pressure correction equation has the following form

$$a_{\mathrm{P}}^{p} p_{\mathrm{P}}' = \sum a_{\mathrm{nb}}^{p} p_{\mathrm{nb}}' + b_{\mathrm{P}}', \tag{21}$$

where the term b_{P}', called the mass residual, in the pressure correction equation is the left-hand side of the discretised continuity equation evaluated in terms of the fluid velocities $\boldsymbol{u}^*$.

In the PISO algorithm [76], a second correction step is introduced as

$$a_{\mathrm{P}}^{p} p_{\mathrm{P}}'' = \sum a_{\mathrm{nb}}^{p} p_{\mathrm{nb}}'' + b_{\mathrm{P}}'', \tag{22}$$

where the coefficients have the same value in the first pressure correction equation shown in Eq. (21) and the source term has been changed based on the value of first velocity correction $\boldsymbol{u}'$.

After solving the first and second pressure corrections, the solutions are updated as

$$\begin{aligned} p &= p^* + p' + p'', \\ \boldsymbol{u} &= \boldsymbol{u}^* + \boldsymbol{u}' + \boldsymbol{u}'', \end{aligned} \tag{23}$$

where

$$\begin{aligned} \boldsymbol{u}'_\mathrm{f} &= \frac{A_\mathrm{f}}{a^{\boldsymbol{u}}_\mathrm{P}}(p'_\mathrm{P} - p'_\mathrm{nb}), \\ \boldsymbol{u}''_\mathrm{f} &= \frac{\sum a^{\boldsymbol{u}}_\mathrm{nb}\boldsymbol{u}'_\mathrm{nb} + A_\mathrm{f}(p''_\mathrm{P} - p''_\mathrm{nb})}{a^{\boldsymbol{u}}_\mathrm{P}}. \end{aligned} \tag{24}$$

3.4. VOF solver for interface capturing

A key requirement for simulating two-phase flows is a method for calculating the shape of the interface. Numerous methods have been proposed and used for the simulation of free-surface/two-phase flows. However, the VOF method for capturing the interface is one of the most popular approaches due to its advantages: mass conservation, computational efficiency and easy implementation. From a general point of view, there are two classes of algorithms to solve the F transport equation (Eq. (4)): algebraic and geometric computation [78]. Excellent reviews on the VOF methods can be found in [78, 79].

Considering the advantages of the VOF method and efficiency in algebraic computation, the high resolution VOF scheme Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) is employed in this study to capture the air-water interface for two-phase flows. CICSAM is a high resolution scheme based on the normalised variable diagram used by Leonard [17]. It contains two high resolution schemes and the weighting factor is based on the angle between the interface and the direction of motion. An outline of CICSAM is given below. Refer to [80] for more details.

The normalised variable $\tilde{F}$ is defined as

$$\tilde{F} = \frac{F - F_{\mathrm{U}}}{F_{\mathrm{A}} - F_{\mathrm{U}}}, \tag{25}$$

where the subscript A indicates the acceptor and U the upwind cell. The Hyper-C scheme [17], which follows the upper bound of the Convection Boundedness Criteria (CBC) is used as it is highly compressive and can convert a smooth gradient into a sharp step.

$$\tilde{F}_{\mathrm{f}_{\mathrm{CBC}}} = \begin{cases} \min\left\{1, \dfrac{\tilde{F}_{\mathrm{D}}}{c_{\mathrm{D}}}\right\}, & \text{when } 0 \leqslant \tilde{F}_{\mathrm{D}} \leqslant 1 \\ \tilde{F}_{\mathrm{D}}, & \text{when } \tilde{F}_{\mathrm{D}} < 0, \tilde{F}_{\mathrm{D}} > 1 \end{cases} \tag{26}$$

where subscript D indicates donor cell, $c_{\mathrm{D}} = \sum_{\mathrm{f}} \max\left\{\frac{-V_{\mathrm{f}}\Delta t}{\Omega_{\mathrm{D}}}, 0\right\}$ is the Courant number of the donor cell and V_{f} is the volumetric flux. However, the Hyper-C scheme is inadequate to preserve the shape of an interface which lies tangentially to the flow direction. Thus CICSAM switches to the ULTIMATE-QUICKEST (UQ) scheme [17]

$$\tilde{F}_{\mathrm{f}_{\mathrm{UQ}}} = \begin{cases} \min\left\{\dfrac{8c_{\mathrm{D}}\tilde{F}_{\mathrm{D}} + (1 - c_{\mathrm{D}})(6\tilde{F}_{\mathrm{D}} + 3)}{8}, \tilde{F}_{\mathrm{f}_{\mathrm{CBC}}}\right\}, & \text{when } 0 \leqslant \tilde{F}_{\mathrm{D}} \leqslant 1 \\ \tilde{F}_{\mathrm{D}}, & \text{when } \tilde{F}_{\mathrm{D}} < 0, \tilde{F}_{\mathrm{D}} > 1 \end{cases} \tag{27}$$

in this case.

Thus, depending on the angle between the interface and the flow, CIC-

SAM combines these two schemes, then

$$\tilde{F}_{\mathrm{f}} = \gamma_{\mathrm{f}} \tilde{F}_{\mathrm{f}_{\mathrm{CBC}}} + (1 - \gamma_{\mathrm{f}}) \tilde{F}_{\mathrm{f}_{\mathrm{UQ}}}, \tag{28}$$

in which the weighting factor is given as

$$\gamma_{\mathrm{f}} = \min\left\{ k_{\gamma} \frac{\cos(2\alpha_{\gamma}) + 1}{2}, 1 \right\}, \tag{29}$$

where k_{γ} is a constant introduced to control the dominance of the different schemes and the recommended value is $k_{\gamma} = 1$, α_{γ} is the angle between the vector normal to the interface and the vector which convects the centres of donor and acceptor cells.

The final expression for the face value of F is

$$F_{\mathrm{f}} = (1 - \beta_{\mathrm{f}}) F_{\mathrm{D}} + \beta_{\mathrm{f}} F_{\mathrm{A}}, \tag{30}$$

where the weight factor β_{f} is obtained by

$$\beta_{\mathrm{f}} = \frac{\tilde{F}_{\mathrm{f}} - \tilde{F}_{\mathrm{D}}}{1 - \tilde{F}_{\mathrm{D}}}. \tag{31}$$

It is noted that the normalised variable in Eq. (25) will be divided by zero if the volume fraction F has the same value in the acceptor and upwind cell. In the numerical implementation, the numerator and denominator of the weighting factor in Eq. (31) are multiplied by $(F_{\mathrm{A}} - F_{\mathrm{U}})$, resulting in a modified expression for the normalised variable on the face (not shown here), in order to avoid the singularity in the computation. In the present study, for cut-cells, the flux across each face of the control volume is modified when solving the volume fraction F equation, which will be discussed in Section 3.5.

3.5. Cartesian cut-cell method

For Cartesian cut-cell methods, an instability problem might occur in small cells when explicit schemes are used. Thus, the cell-merging technique [81] as well as slightly modifying control volumes [66] are usually employed to avoid numerical instability, both of which effectively increase the size of the cut cell. As an alternative, flux-redistribution schemes [57–59] may be employed to deal with small cut-cells. This problem is very cumbersome especially for moving body problems as cut-cells require updating/modification in every time step. In the present study, the cut cells are not modified and it is found that numerical instabilities are absent due to the implicit scheme used for time integration.

3.5.1. Cut-cell information

In the present study, the geometry of the solid is represented by a general level set function $LS(x, y, z, t)$, in which the boundary of the moving body is represented by $LS(x, y, z, t) = 0$ while the fluid domain has the value of $LS(x, y, z, t) > 0$ and the solid domain is $LS(x, y, z, t) < 0$. The cut-cell interface between the fluid and solid is calculated by a piecewise linear interface (as shown in Fig. 1), which is a straight-line in 2D and a plane in 3D.

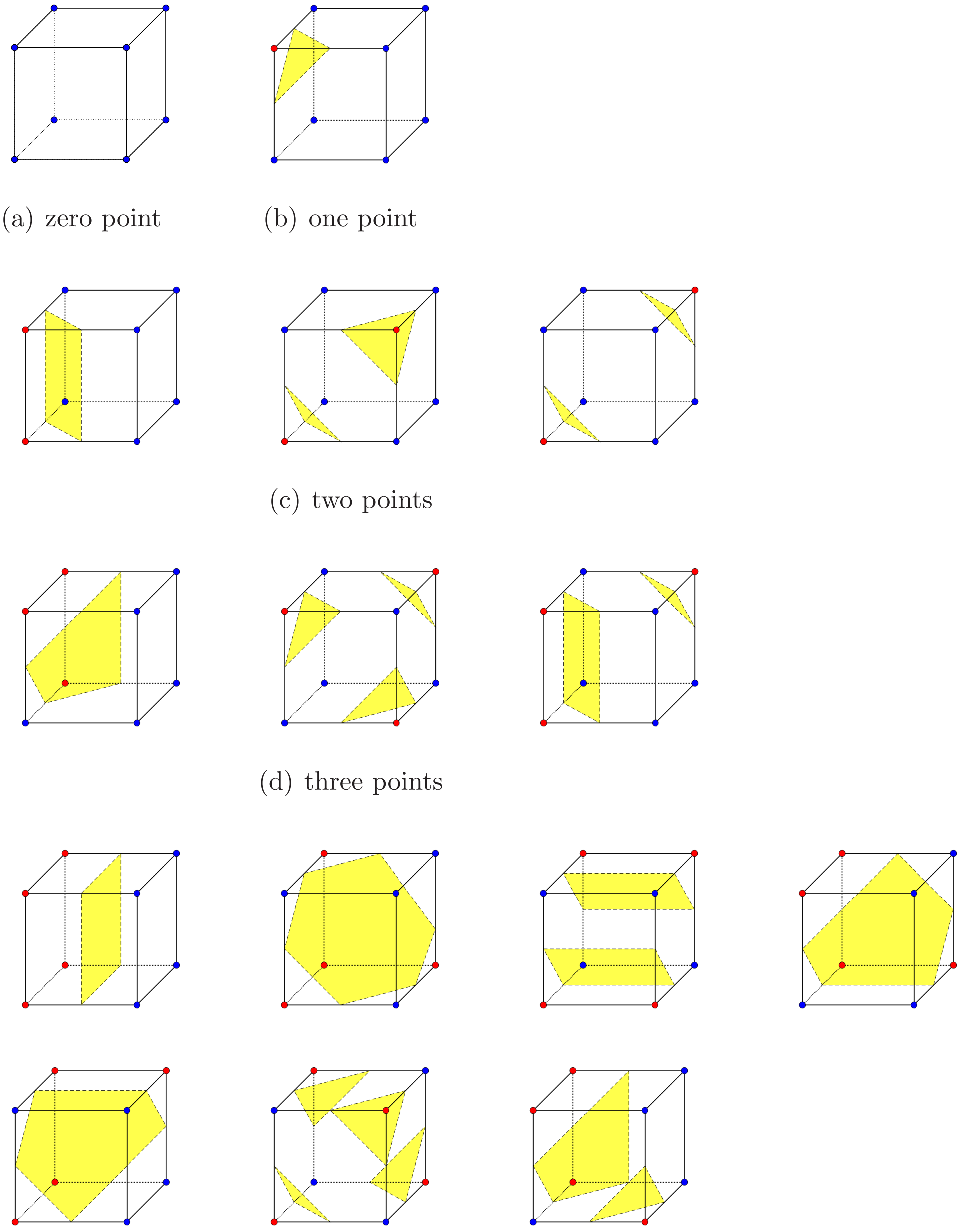
For each 3D Cartesian grid cell, the area of each surface and the total volume available for fluids needs to be calculated. A θ function is defined here, the value of which is 1 for a point accessible to fluid and 0 for a point inside a solid. The average of this function over a control volume or cell face is the fraction of the volume or area available to the flow. By using this approach, the volume of the solid can be obtained as:

$$V_{\text{solid}}(t) = (1 - \theta(t))V. \tag{32}$$

It is worth noting that θ is a function of time t here, for a stationary body, the θ function is only needed to be calculated once at the beginning as used for a fixed body simulation.

In order to calculate the area and volume of the cut-cell, how the solid boundary cuts the Cartesian cell needs to be determined first. The marching tube algorithm [82] is employed and the grid nodes can be classified as inside or outside of the solid based on the sign of the level set function $LS(x, y, z, t)$. For 3D, there are 8 grid nodes in each cell so that 2^8 configurations are possible. By using the inversion and rotation of different configurations, the 256 cases can be reduced to 15 cases as shown in Fig. 3 for a typical cut cell in a 3D Cartesian grid, which includes zero, one, two, three, and four points being cut through the Cartesian grid cell.

Once the configuration of the cut cell has been determined, the intersection point along each edge can be obtained by linear interpolation of the level set function of two neighbouring grid nodes. Once the point of intersection of the line with the cut plane is found, the geometric information can be calculated. Finally, the face areas and total volume of the truncated cell are calculated by numerical integration. In contrast to a full fluid cell, the spatial discretisation at cell faces and cell centres are modified in a cut cell, which will be presented in the following terms.



(a) zero point

(b) one point

(c) two points

(d) three points

(e) four points

Figure 3: Example of 3D cut cell configurations, which show how many points have been cut through a Cartesian grid cell: (a) zero point (either pure fluid or solid cell); (b) one point; (c) two points; (d) three points; (e) four points. There are 256 cases in total, but after inversion and rotation, the 256 cases can be reduced to 15 cases shown here.

3.5.2. Advection term

The finite volume discretisation of the advection term in Eq. (13) is obtained as

$$
\begin{aligned}
Q_\varphi^{\mathrm{C}} = \iint_S (\rho \boldsymbol{u} \cdot \boldsymbol{n}) \varphi \mathrm{d}S &= \sum_{\mathrm{f}} (\theta A)_{\mathrm{f}} \left[(\rho \boldsymbol{u} \cdot \boldsymbol{n}) \varphi \right]_{\mathrm{f}} \\
&= \sum_{\mathrm{f}} (\rho \boldsymbol{u} \cdot \boldsymbol{n} \theta A)_{\mathrm{f}} \varphi_{\mathrm{f}} \\
&= \sum_{\mathrm{f}} m_{\mathrm{f}} \varphi_{\mathrm{f}},
\end{aligned}
\tag{33}
$$

where the subscript f denotes the corresponding face of the control volume, A is the area of the face and m is the mass flux through the face

$$
m = \rho \boldsymbol{u} \cdot \boldsymbol{n} \theta A. \tag{34}
$$

In cut cells, the mass flux has also to be modified by the θ function on the boundary. If $\theta = 0$ (such as the west face of the first case in Fig. 3(e)), there is no mass flux through the face and the advective flux is obtained as

$$
m_{\mathrm{f}} = 0. \tag{35}
$$

The mass flux at the faces of the momentum control volume can be obtained by the interpolation of values of ρ and $\boldsymbol{u}$, such as $m_{\mathrm{f}} = \rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}} \cdot \boldsymbol{n} \theta_{\mathrm{f}} A_{\mathrm{f}}$, however, the mass conservation in the momentum control volume can be only guaranteed to the accuracy of the interpolation procedure [18]. Thus, in this study, the m_{f} is obtained from the interpolation of the mass fluxes, which is already available at the faces of the continuity control volumes.

For interface capturing, the face value F_{f} is obtained from the CICSAM scheme in Eq. (31). For the momentum equations, the face value φ_{f} can be obtained from different schemes and more details can be found in [74]. A high resolution scheme [73], which combines the high-order accuracy with monotonicity, is used in this study to discretise the value at the face as

$$\varphi_{\mathrm{f}}{}^{\mathrm{HRS}} = \varphi_{\mathrm{f}}^{\mathrm{FOU}} + \Phi(r_{\mathrm{f}}^{m})\Psi(r_{\mathrm{f}}^{\varphi})(\varphi_{\mathrm{f}}^{\mathrm{SOU}} - \varphi_{\mathrm{f}}^{\mathrm{FOU}}), \tag{36}$$

in which $\varphi_{\mathrm{f}}^{\mathrm{FOU}}$ and $\varphi_{\mathrm{f}}^{\mathrm{SOU}}$ are the corresponding values obtained from the first-order and second-order upwind schemes.

$\Psi()$ is the limiter function where the minmod limiter [83], which is one of the simplest second-order TVD (Total Variation Diminishing) schemes, is used here

$$\Psi(r) = \max[0, \min(r, 1)]. \tag{37}$$

Similar results are obtained by using other limiter functions and a general review of various limiter functions can be found in [84] and will not be repeated here. r_{f}^{φ} represents the ratio of successive gradients of φ on the solution mesh and obtained as

$$r_{\mathrm{f}}^{\varphi} = \frac{\varphi_{\mathrm{A}} - \varphi_{\mathrm{D}}}{\varphi_{\mathrm{D}} - \varphi_{\mathrm{U}}}, \tag{38}$$

where subscripts D, A, and U denote donor cell, acceptor cell, and upwind cell, respectively.

As discussed in the previous study [74], $\Phi()$ is the step function and r_{f}^{m} is the variation for the mass, defined as the ratio of the mass flux between the conservative and non-conservative forms

$$r_{\mathrm{f}}^{m} = \frac{(\rho \boldsymbol{u} \cdot \boldsymbol{n} \theta A)_{\mathrm{f}}}{\rho_{\mathrm{c}}(\boldsymbol{u} \cdot \boldsymbol{n} \theta A)_{\mathrm{f}}}. \tag{39}$$

The step function $\Phi()$ takes the form

$$\Phi(r) = \begin{cases} 1, & \text{if } |r| \leqslant 1; \\ 0, & \text{otherwise}, \end{cases} \tag{40}$$

which means that the present high resolution scheme switches to the first-order upwind scheme when the density on the cell face is larger than the density in the cell centre.

3.5.3. Diffusion term

The finite volume discretisation of the diffusion term in Eq. (13) is obtained as

$$Q_{\varphi}^{\mathrm{D}} = \iint_{S} \Gamma \frac{\partial \varphi}{\partial n} \mathrm{d}S = \sum_{\mathrm{f}} \Gamma_{\mathrm{f}} \frac{\partial \varphi}{\partial n} (\theta A)_{\mathrm{f}}, \tag{41}$$

where the viscosity on the face is obtained by the harmonic mean [85], for example, on the east face

$$\Gamma_{\mathrm{e}} = \frac{\Gamma_{\mathrm{P}} \Gamma_{\mathrm{E}}}{\lambda_{\mathrm{e}} \Gamma_{\mathrm{P}} + (1 - \lambda_{\mathrm{e}}) \Gamma_{\mathrm{E}}}, \tag{42}$$

where $\lambda_{\mathrm{e}} = |\mathrm{eE}| \, / \, |\mathrm{PE}|$. Analogous expressions can be derived for all other faces (f=w, n, s, b, r) by making appropriate index substitutions and will not be shown here.

The gradient at the face is calculated by the finite difference approach as

$$\frac{\partial \varphi}{\partial n} = \frac{\varphi_{\mathrm{nb}} - \varphi_{\mathrm{P}}}{\Delta_{\mathrm{Pnb}}}, \tag{43}$$

where Δ_{Pnb} is the distance from the present point P to the neighbouring point nb.

When the control volume is a cut cell, special attention has to be paid to the spatial discretisation. When the face of a momentum control volume is on the wall, the diffusion flux is obtained as

$$\iint_S \Gamma \frac{\partial \varphi}{\partial n} \mathrm{d}S = \sum_{\mathrm{f}} \Gamma_{\mathrm{f}} \frac{\partial \varphi}{\partial n} (\theta A)_{\mathrm{f}} + \tau_{\mathrm{w}}[(1-\theta)A]_{\mathrm{f}}, \tag{44}$$

where $\frac{\partial \varphi}{\partial n}$ is calculated by the finite difference approach in (43) and τ_{w} is the shear stress of the wall on the face of the control volume.

3.5.4. Source term

The finite volume discretisation of the source term in Eq. (13) is obtained as below:

Pressure term. The finite volume discretisation of the pressure term is obtained as

$$Q_{\varphi}^{\mathrm{p}} = \iiint_{\Omega} -\boldsymbol{\nabla} p \mathrm{d}\Omega = -\boldsymbol{\nabla} p \theta_{\mathrm{c}} \Omega, \tag{45}$$

and the pressure gradient is calculated as

$$\begin{aligned} \boldsymbol{\nabla} p &= (\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z}) \\ &= (\frac{p_{\mathrm{e}} - p_{\mathrm{w}}}{\Delta x}, \frac{p_{\mathrm{b}} - p_{\mathrm{r}}}{\Delta y}, \frac{p_{\mathrm{n}} - p_{\mathrm{s}}}{\Delta z}). \end{aligned} \tag{46}$$

Body force term. The finite volume discretisation of the body force term is obtained as

$$Q_{\varphi}^{\mathrm{B}} = \iiint_{\Omega} \rho \boldsymbol{g} \mathrm{d}\Omega = \rho_{\mathrm{c}} \boldsymbol{g} \theta_{\mathrm{c}} \Omega, \tag{47}$$

where the value in the centre of the control volume is obtained by the volume averaging of two values on the face of the control volume.

Internal source term. The finite volume discretisation of the internal source term is obtained as

$$Q_{\varphi}^{\mathrm{I}} = \iiint_{\Omega} \rho \boldsymbol{u} \psi(t) \mathrm{d}\Omega = \rho_{\mathrm{c}} \boldsymbol{u} \psi_{\mathrm{c}}(t) \theta_{\mathrm{c}} \Omega. \tag{48}$$

3.6. Procedure

In summary, the overall solution procedure of the present two-phase flow model with moving bodies for one time step is detailed as:

(1) Setup boundary conditions and moving bodies with pre-specified motion: Section 2.5.

(2) Calculate the cut-cell information and the θ function for the area and volume available for the flow in each cells, and compute the internal source function $\psi(t)$: Section 3.5.1.

(3) Solve the volume-of-fluid function F: Section 3.4.

(4) Update the fluid properties, density ρ and dynamic viscosity μ: Section 2.2.2.

(5) Calculate the turbulent eddy viscosity μ_t if the dynamic SGS model is used: Section 2.4.

(6) Setup the coefficients $a^{\boldsymbol{u}}$ and $b_{\mathrm{P}}^{\boldsymbol{u}}$ based on the latest pressure p^* and velocity $\boldsymbol{u}$ and solve for the intermediate velocity $\boldsymbol{u}^*$: Section 3.3.3.

(7) Setup the coefficients a^p and b'_{P} and solve for the first pressure correction p' and calculate $\boldsymbol{u}'$: Section 3.3.3.

(8) Setup the coefficients b''_{P} and solve for the second pressure correction p'' and calculate $\boldsymbol{u}''$: Section 3.3.3.

(9) Update the pressure $p = p^* + p' + p''$ and the velocity $\boldsymbol{u}^* + \boldsymbol{u}' + \boldsymbol{u}''$: Section 3.3.3.

(10) Go back to step (6) if the mass residule is larger than a threshold, otherwise go to next time level in step (1): Section 3.3.

4. Results and discussion

A grid convergence study of the two-phase flow model, the benchmark problem of a 3D collapse of a water column without a structure is carried out first to test the interface capturing method at various grid resolutions. The validation of the cut-cell method is then demonstrated by studying dambreak flow interacting with a fixed square cylinder. Numerical results are quantitatively compared with experimental data in terms of velocity and force acting on the cylinder. With the goal to validate the cut-cell method involving moving bodies, a single-phase flow problem for a moving cylinder in a quiescent fluid is studied followed by a two-phase flow with moving bodies. Therefore, 2D dambreak flow over a wet bed is studied with the movement of the gate being simulated directly using the proposed moving cut-cell method, and the computed water surface behavior is compared qualitatively with experimental observations. After that, the validation of the Cartesian cut-cell/VOF method with moving bodies is further demonstrated by studying the exit and entry of a circular cylinder into a fluid. Finally, the method is employed to study 3D water wave generation due to a sliding wedge.

4.1. Convergence study - 3D collapse of a water column

The classical benchmark case of a 3D dambreak flow is considered, which has been studied in detail experimentally in [86] and numerically, for instance in [87]. The motion of the water (i.e. water height on the wall and spread length on the bottom) during water collapse were recorded in the experiment and are used here to validate the present two-phase flow model.

The computational domain is $4a \times a \times 4a$ in the streamwise, spanwise, and vertical directions, respectively. Initially, a column of water with $2a$ height and a width is placed on the bottom left of the tank and $a = 0.25$ m is used here. Here, b and c denote the time history of the remaining height and surge front position of the water column and the schematic is shown in Fig. 4. The computational domain is discretised with three different sets of uniform meshes with $32 \times 8 \times 32$, $64 \times 16 \times 64$, and $128 \times 32 \times 128$ in the streamwise, spanwise, and vertical directions, respectively.

Fig. 4 depicts snapshots of the remaining water height (left) and surge fron position (right) as computed together with experimental data [86]. As far as the water column height is concerned, simulation results on all three grids are in very good agreement with the experimental data (Fig. 4, left). As far as the surge front position is concerned, due to the fact that the dam cannot be removed instantaneously in the experiment, a small time

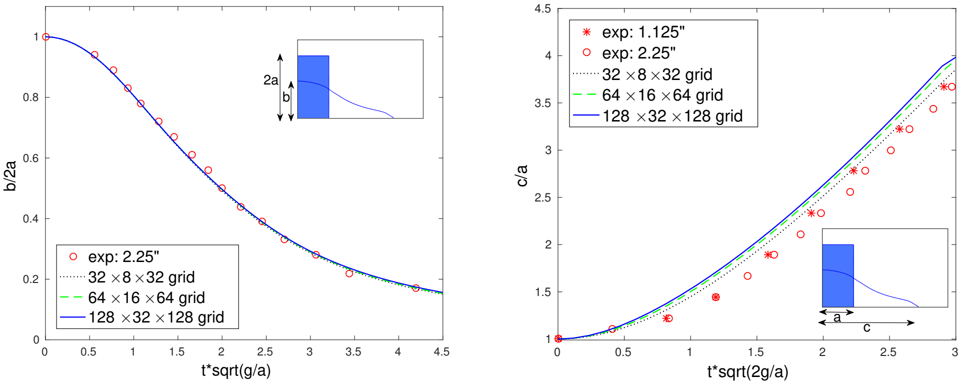


Figure 4: Results of the numerical simulations on three different meshes together with experimental data [86] for remaining water column height b (left) and surge front position c (right), both as a function non-dimensional time.

lag in the experimental data should exist, which is also observed in other numerical simulations. In general, the numerical results converge to the correct solution when refining the mesh and there is only a marginal difference in surge position between the medium and fine mesh simulations.

In order to further study the convergence rate of the present method, the free surface profiles obtained by the simulations on the three meshes at an instant $t = 0.2$ s (shown in the schematic of Fig. 4) are compared with the benchmark solution. An additional simulation with an even finer mesh $256 \times 64 \times 256$ is carried out until $t = 0.2$ s and the so-computed air-water interface is taken as the benchmark solution. Fig. 5 shows the calculated L_1, L_2, and L_{inf} errors with respect to the benchmark solution. It can be seen that the convergence rate is between first-order and second-order, and it is more close to second-order when approaching the fine mesh. This is expected as although second-order discretisation is used for the diffusion term, the

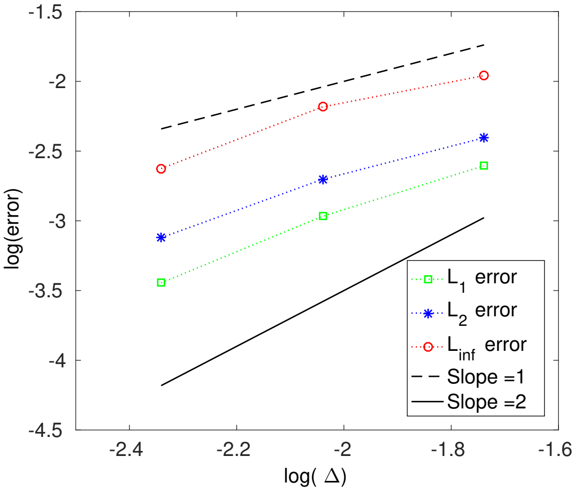


Figure 5: Error norm as a function of grid resolution for the 3D collapse of a water column. Lines for first-order and second-order behavior are also plotted for reference.

advection term is discretised by a high-resolution scheme, which combines the high order accuracy with monotonicity for the first-order scheme.

4.2. 3D dambreak flow interacting with a fixed square cylinder

In this section, a 3D two-phase flow interacting with a fixed structure is considered to verify and validate the implementation of the Cartesian cut-cell method. The dambreak flow interacting with a square cylinder is often considered as a benchmark case and this has been studied in [70], among others. The computational setup (shown in Fig. 6) is the same as in [70] except that a full domain is considered in the present study, rather than using only half of the domain with symmetry boundary conditions along the centre of the domain as was used by [70]. The tank dimensions are 1.6 m × 0.61 m × 0.60 m in the streamwise, spanwise, and vertical direction, respectively. The initial height of the water behind a thin gate is 0.4 m and a

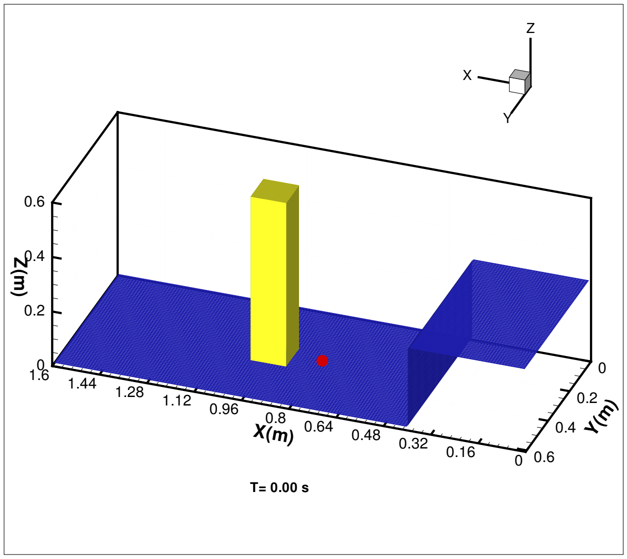


Figure 6: Computational setup for the 3D dambreak flow over a square cylinder. The red dot indicates the location where velocity measurements were carried out.

thin layer of water of 0.01 m depth is set downstream of the gate. The square cylinder (with 0.12 m edge) is placed 0.5 m downstream of the gate and the lateral distance to the sidewalls is 0.24 m. In the experiments, the net force on the cylinder was measured and an LDV system was employed to monitor the fluid velocity at a fixed location (0.146 m upstream of the centre of the cylinder and 0.026 m off the floor of the tank) indicated as a red point in Fig. 6. The computational domain is discretised with a $160 \times 61 \times 60$ uniform mesh, where $\Delta x = \Delta y = \Delta z = 0.01$ m, the same as used in [70].

Fig. 7 presents snapshots of the air-water interface as the flow interacts with the square cylinder. Initially, the water column collapses due to gravity and a two-dimensional bore is generated. When the bore hits the cylinder, three-dimensional flow is observed. Part of the bore reflects back in front of

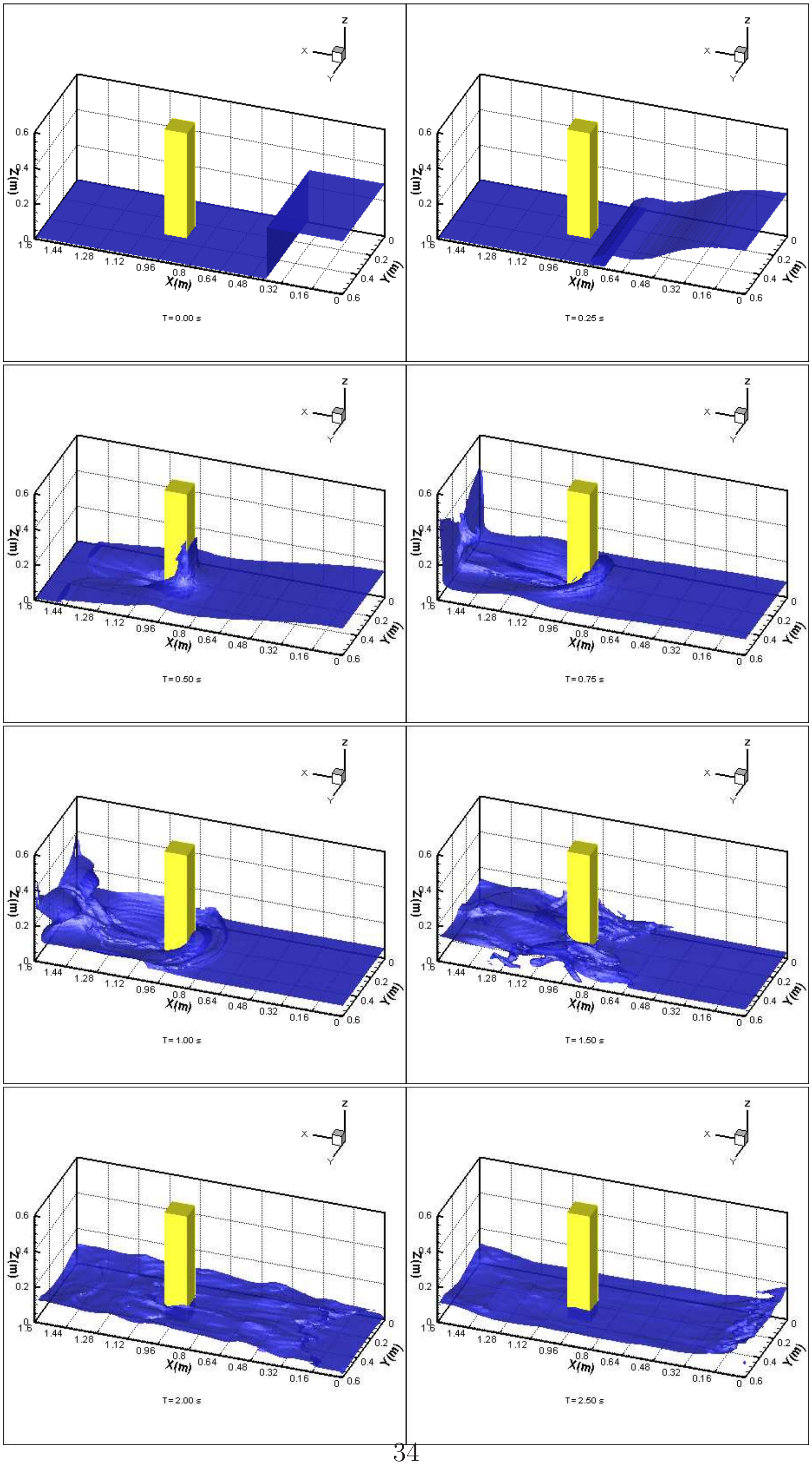


Figure 7: Snapshots of the air-water interface at selected instants in time

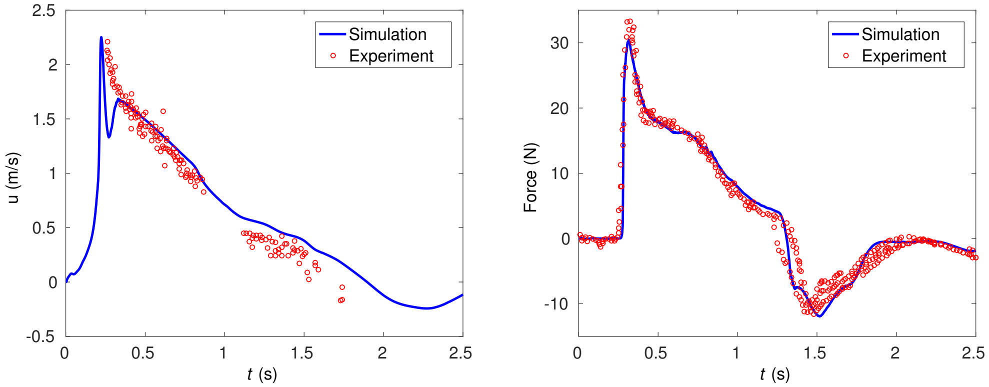


Figure 8: Time-history of horizontal velocity (left) and force acting on the square cylinder (right). The location where velocities are measured is indicated as red point in Fig. 6).

the cylinder while other parts wrap around the cylinder. The separated two bores merge together in the wake zone downstream and start to climb on the end wall. After that, the main bore collapses and is reflected back from the end wall to catch up with the reflected bore from the cylinder, generating splash-up and air entrainment. These snapshots qualitatively agree well with the simulation results in [70].

The fluid velocity and net force acting on the cylinder as a function of time are plotted in Fig. 8. Overall, the numerical results (solid lines) agree well with the experimental data (symbol), especially peak value of horizontal velocity and phase of the force acting on the square cylinder, which demonstrates the accuracy of the Cartesian cut-cell method in the two-phase flow model.

4.3. Moving circular cylinder in a quiescent fluid

After validating the two-phase flow model interacting with a fixed body, moving body problems are considered with the goal to validate the moving body algorithm with the cut-cell method. First, a single-phase flow problem of a moving circular cylinder in a quiescent fluid is studied. At each time step the solid volume (Eq. (7)) and hence the cut-cells require updating as the cylinder changes its position whilst fluid movement is computed in the entire domain. The computational domain is $[-10D, 30D] \times [-20D, 20D]$ with a uniform mesh of 1280×1280. The cylinder is initially positioned at $x = 20D$ and suddenly set into motion with a constant velocity of $-U_{\text{mov}}$ moving to the left until $x = 0D$. Zero velocity fluid is set for the initial condition and the no-slip boundary condition is applied at all boundaries. A Reynolds number of $Re = U_{\text{mov}}D/\nu = 40$ is considered. A constant time step with CFL number of 1.0 is used in the simulation in order to demonstrate the capability of the implicit time integration treatment for the cut-cell method.

Fig. 9 shows snapshots of vorticity contours around the impulsively moving circular cylinder at non-dimensional time of 1, 5, 10, and 20. A symmetric vortex pair develops during the movement of the cylinder. The flow reaches a steady state and no vortex shedding occurs at this Reynolds number, in analogy to many other studies of constant flow past a stationary cylinder at $Re = 40$. The drag coefficient C_D as a function of time is plotted in Fig. 10 and the value obtained at the final stage is compared with other experimental and computational studies for fixed and moving cylinders in Table 2. It is worth mentioning that although we use a large CFL number (1.0) and the non-dimensional computational time is shorter than the case for fixed cylinder, our simulation compares well with other studies.

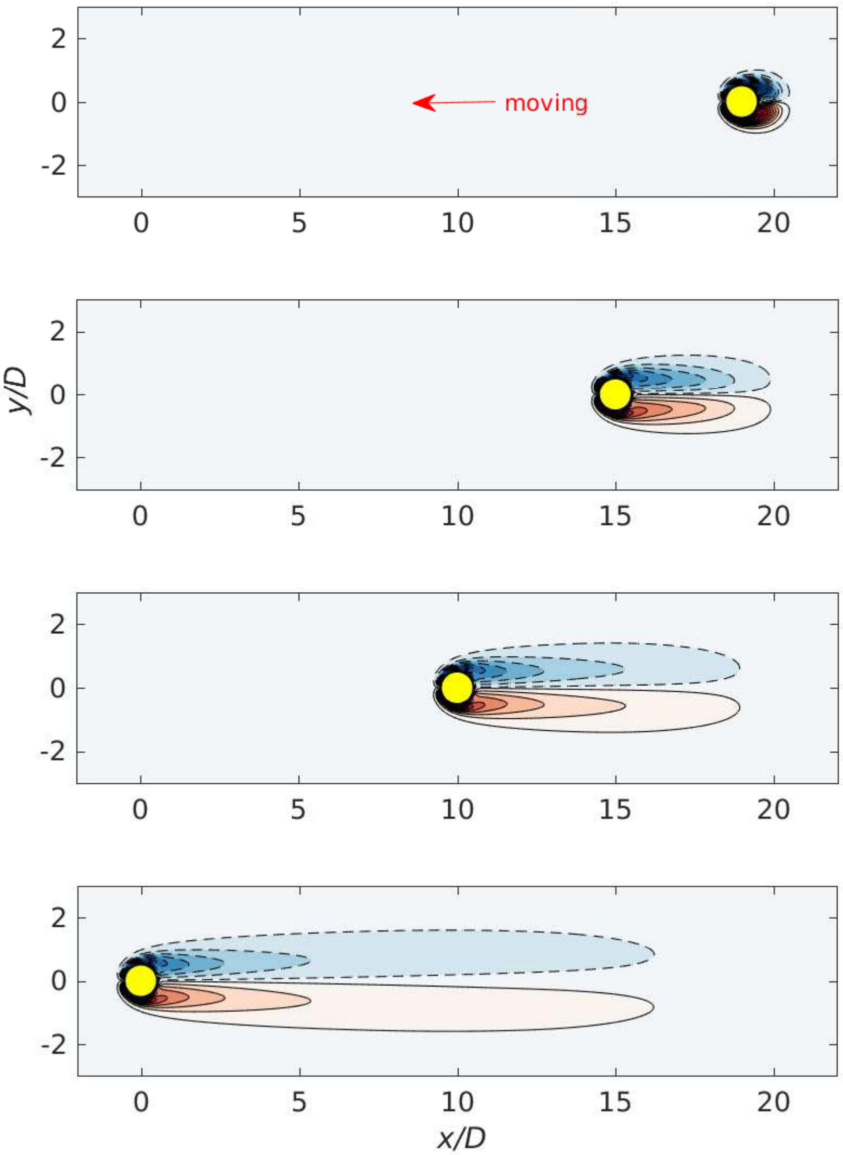


Figure 9: Snapshots of contours of the vorticity around a moving circular cylinder for $Re = 40$ at non-dimensional times of 1, 5, 10, and 20.

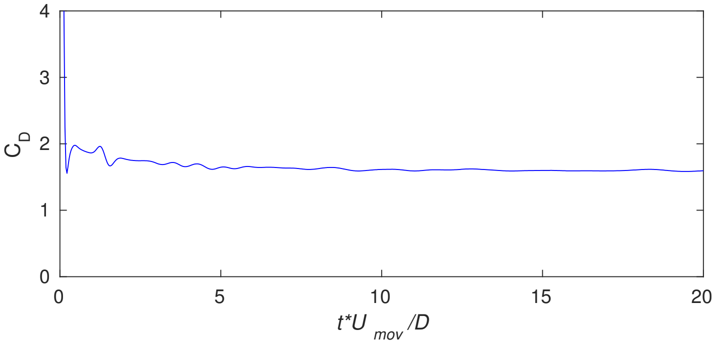


Figure 10: Drag coefficient of a circular cylinder moving through a quiescent fluid at $Re = 40$.

Table 2: Comparison between the experiment, other simulations, and the present study for a cylinder at $Re = 40$.

Study	method	C_D
Tritton [88]	experiment	1.59
Fornberg [89]	fixed cylinder	1.50
Marella et al. [90]	fixed cylinder	1.52
Shirgaonkar et al. [91]	moving cylinder	1.52
Wu et al. [92]	moving cylinder	1.554
Present	moving cylinder	1.58

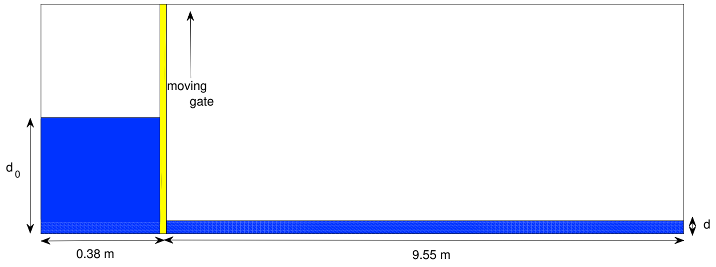


Figure 11: Computational setup for the 2D dambreak flow over a wet bed (not to scale). The gate is moving with a constant vertical velocity $U_{\text{gate}} = 1.5$ m/s.

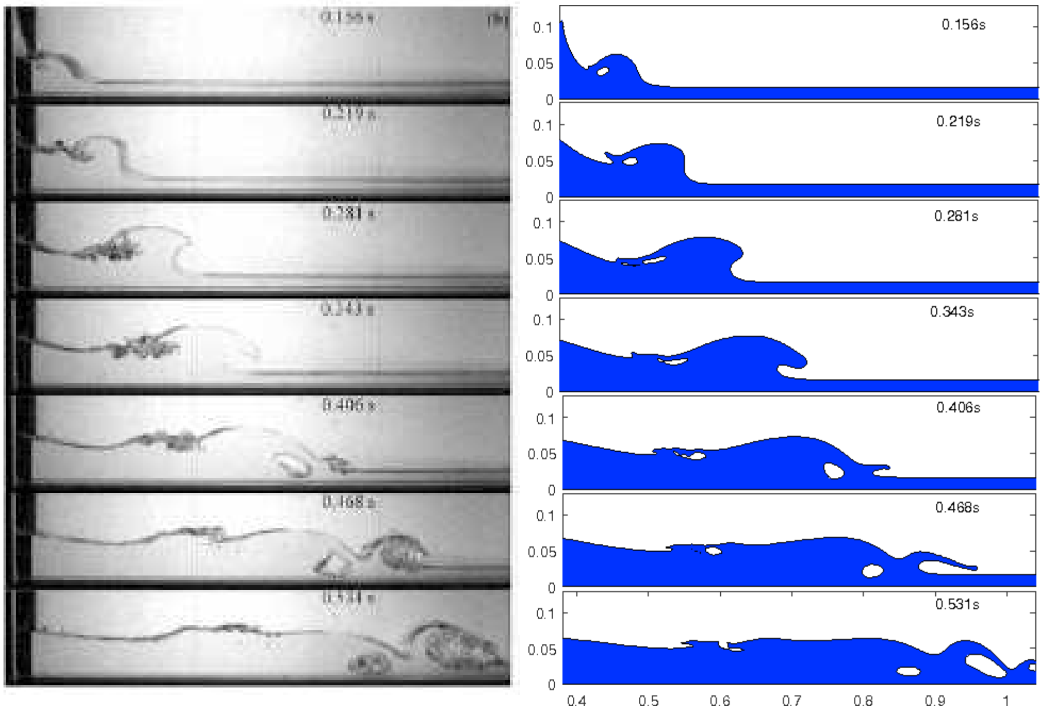
4.4. Dambreak over a wet bed with a moving gate

After successful validation of the moving body algorithm using the cut-cell method for single-phase flows, moving body problems for two-phase flows are considered. The dambreak flow over a wet bed with a moving lock gate is studied using experimental data reported in [93]. In the experiment, the tank has two parts with a lock gate separating a channel and a lock as shown in Fig. 11. The length of the channel downstream is 9.55 m covered by 0.018 m deep water and the length of the lock upstream is 0.38 m and its initial height of water is 0.15 m. In the experiment, the gate separating the lock and the channel was moved upward with an approximate constant velocity of $U_{\text{gate}} = 1.5$ m/s. The computational domain of dimensions 9.93 m × 0.3 m is discretised by a 800 × 80 non-uniform mesh in the streamwise, vertical directions, respectively. The fine mesh of $\Delta x = \Delta y = 0.0025$ m is deployed near the gate at the lower part of the computational domain.

The calculated progression of the dambreak flow over a wet bed is visualised by the blue contours indicating the water face in the right column of Fig. 12. Also depicted in the left column are photographs of the experiment of [93] at the same instants in time as the numerical simulation. Due to gravity, the water in the lock (left of the gate) pushes the still water on the wet bed away after removal of the gate. The simulation produces a "mushroom"-shaped jet initially, which agrees well with the experiment. The direct simulation of the moving gate plays an important role here, because some fluid is dragged upwards with the gate and the initial release of the water is restricted and somewhat delayed. Later, the "mushroom" jet develops into a plunging jet, touching down on the water surface and generating a secondary jet with an air cavity underneath; simultaneously the remains of the"mushroom" jet impinge on the water surface on the upstream side of the front with air entrainment during the breaking process. Overall, a good qualitative match is obtained between the simulation and experiment in terms of behaviour of the fluid, capturing well the breaking waves, splash-up and air entrainment.

4.5. Water exit and entry of a circular cylinder

Water exit and entry of solid objects have both theoretical and practical applications, especially in ocean and offshore engineering. Such fluid-structure interaction flows have been studied numerically using the boundary element method (BEM) [94], 2D single-phase free surface models with partial cell treatment [32] and 3D two-phase flow models with the immersed boundary method [33]. The same parameters as those used in [32] are chosen here: the radius of a circular cylinder is $R = 1.0$ m and its centre is initially located at a distance $d = 1.25$ m below the still water level. The gravitational acceleration is $g = 1.0$ m/s^2 and the upward velocity of the cylinder is $U_{\text{cylinder}} = 0.39$ m/s. As the focus is on the impact region, a



(a) experiment

(b) simulation

Figure 12: Visualised dambreak flow as observed in the experiment [93] (a) and as numerically predicted (b). The gate is simulated with a constant upward velocity $U_{\text{gate}} = 1.5$ m/s.

smaller computational domain $20R \times 10R$ is used and it is discretised by a uniform mesh 400×200 in the streamwise and vertical directions, respectively. The mesh resolution $\Delta x = \Delta y = 0.05R$ is the same as in [32, 33] for the fine mesh region in a non-uniform grid. The simulation is carried out until the non-dimensional time $T = |U_{\text{cylinder}} t/D| = 3$.

Fig. 13 shows snapshots of the air-water interface together with the cylinder for the water exit problem. As the cylinder reaches $T = 0.2$, the air-water interface deforms in the region above the cylinder. At $T = 0.4$ and $T = 0.6$, more significant water surface deformation is obeserved taking the shape of the cylinder in its vicinity. At these two time instants, the predicted results compare well with the BEM results by [94] (red symbols). At $T = 0.8$ and $T = 1.0$, a thin layer of fluid is found on the surface of the cylinder and as displaced fluid flows back into the water body a depression of the water surface is observed on either side of the cylinder. The thin layer of fluid around the cylinder eventually dries up ($T = 2.0$) before the cylinder exists the water completely ($T = 3.0$). The present results are similar to those obtained by the single-phase flow model of [32] and very close to those given by the two-phase flow model of [33]. The thin water film over the cylinder appears better resolved with two-phase flow models.

For the water entry problem, the overall setup is the same as the water exist problem except that the cylinder starts at a distance $d = 1.25$ m above the still water level and moves with a constant downward velocity $U_{\text{cylinder}} = -1.0$ m/s.

Fig. 14 shows profiles of the air-water interface at selected instants in time for water entry of a moving cylinder. Two oblique jets are generated

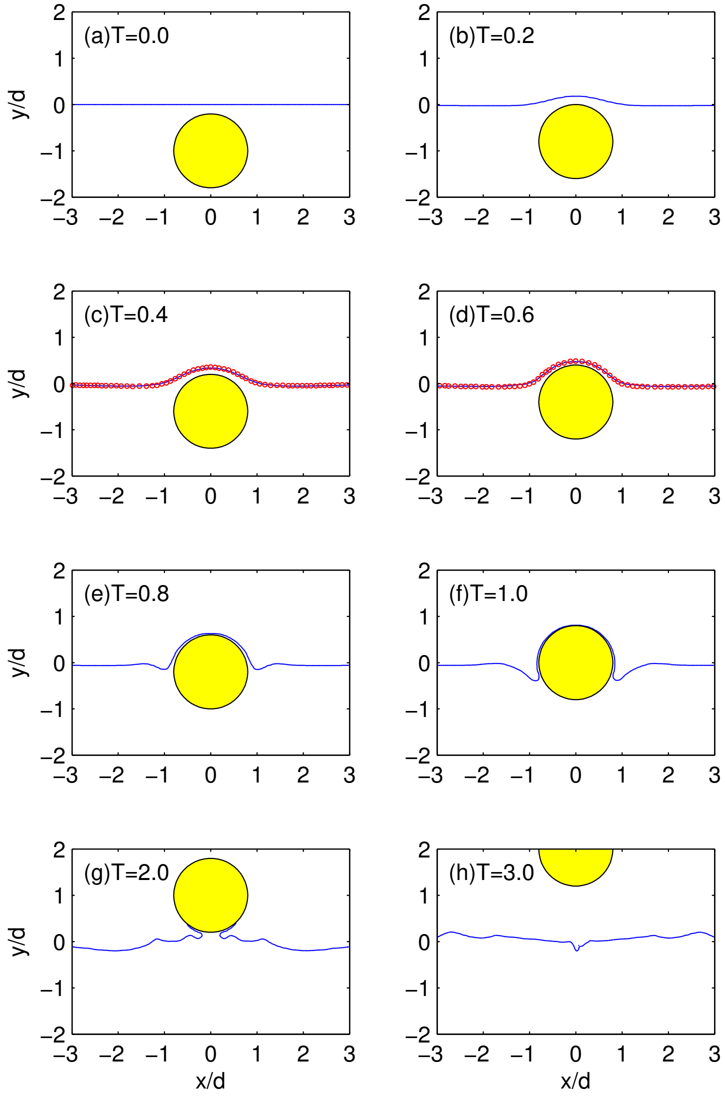


Figure 13: Snapshots of the air-water interface profile (blue line) for water exit of a horizontal cylinder. Also plotted are the BEM results of [94] in (c) and (d).

during the impact and the triple point between air-water-solid moves along the cylinder surface. There is a water surface depression when the cylinder moves further downward. After the cylinder is completely submerged, the elevated areas of water meet in the centre above the cylinder and generate a small upward jet. The simulation results agree with those given in [32] albeit a slightly different jet shape occurs, probably due to the different approach used here.

4.6. Water wave generation by a 3D sliding wedge

The water waves generated by a 3D sliding wedge has been studied both experimentally and numerically in [95]. In the experiments, the initial position of the wedge varied from subaerial to submerged. The subaerial case has been studied in [33, 95] and here the focus is on the submerged wedge case. The setup is the same as in the experiment and details can be found in [95]. The computational domain of 6.6 m × 3.7 m × 3.3 m is discretised by a 220 × 160 × 220 uniform Cartesian grid in the streamwise, spanwise, and vertical directions, respectively, with its origin being setup at the shoreline along the central plane. The motion of the sliding wedge in the simulation is prescribed and is the same as in the experiment.

Fig. 15 shows snapshot of the air-water interface profile together with the location of the wedge as it moves down the slope. At the beginning, the submerged wedge induces downward velocity above the wedge, which causes a depressed air-water interface above the wedge and push the water surface upward in front of the wedge. A small positive wave and a large negative wave are generated as the wedge reaches its full speed. A positive wave is also generated behind the wedge when it moves down. It can be seen that

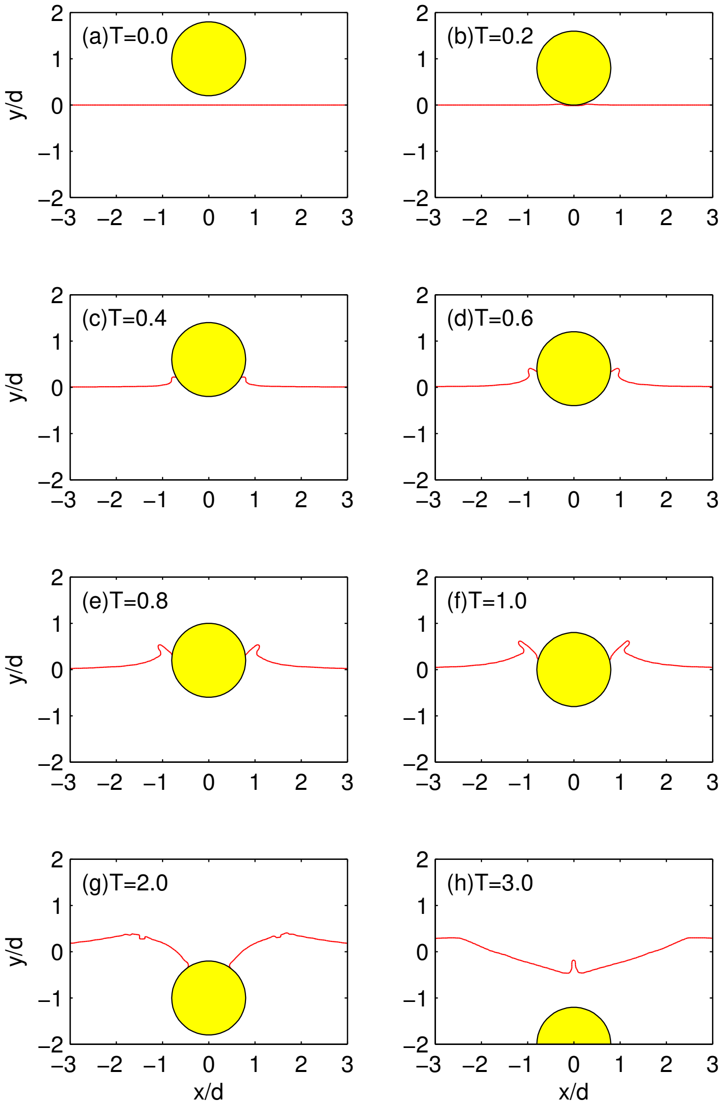


Figure 14: Profiles of the air-water interface for water entry of a horizontal cylinder at slected instants in time.

three-dimensional waves are generated due to the 3D wedge, propagating laterally offshore. In the shoreline region, because of the slide movement, the runup height has negative values at the beginning and a rebounding wave causes the shoreline to rise up later and propagate outward.

In order to quantitatively assess the simulation's accuracy, the time history of the computed water surface elevation and the wave gauge data of the experiment are plotted in Fig. 16 for comparison. The two wave gauges (one in front of the wedge's initial location and the other is at the side of the wedge's initial location shown in Fig. 15). The overall agreement between simulation and measurement is quite convincing, especially at the beginning of the sliding process.

5. Conclusions

In this paper, a Cartesian cut-cell/volume-of-fluid method has been developed for the computation of 3D two-phase flows with moving bodies. The method is based on the 3D two-phase flow model over fixed complex topography of [96] to which a moving body algorithm with the source function approach [69, 70] has been added. The method of large-eddy simulation has been adopted, i.e. it is solves the space filtered Navier–Stokes equations and employs the dynamic Smagorinsky subgrid-scale model to compute the unresolved scales of turbulence. The finite volume method has been adopted to discretise spatial derivatives with the PISO algorithm for the pressure-velocity coupling, and a backward finite difference discretisation to integrate the equations in time, leading to a fully implicit scheme for the governing equations. The air-water interface is captured using the high resolution

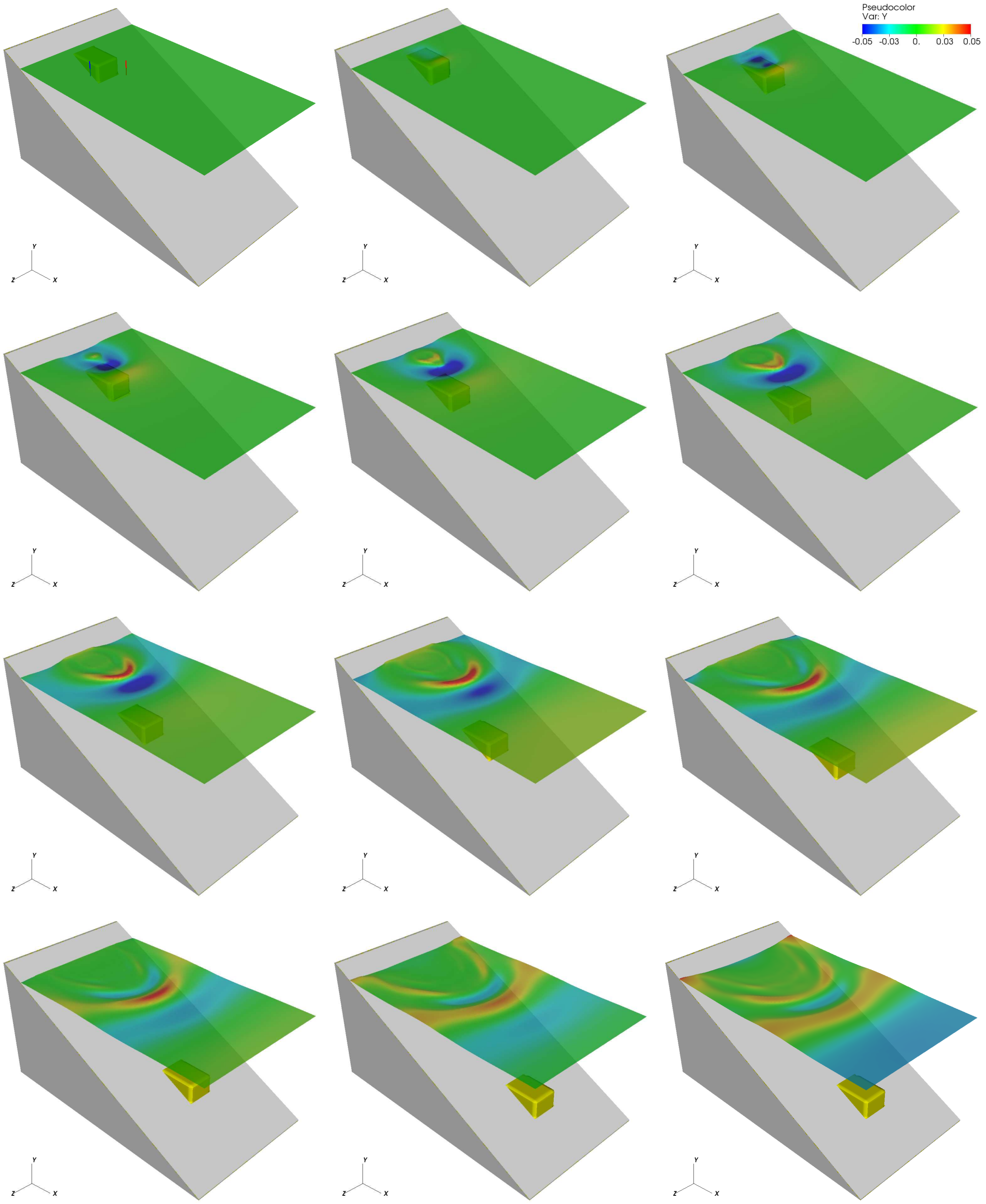
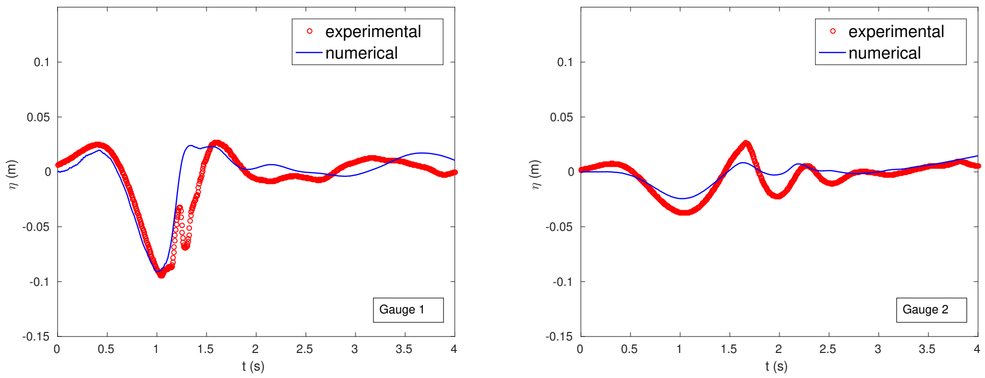


Figure 15: Air-water interface at selected instants in time. The wedge's initial position is 0.1 m below the mean water level. The time interval between plots is 0.3 s. The two wave gauges are located at $(x, z) = (1.83, 0)$ and $(x, z) = (1.2446, 0.635)$ shown as vertical lines on the top left figure.



(a) (b)

Figure 16: Computed and measured time history of the water surface elevation at two wave gauges. The location of wave gauges (a) and (b) are at $(x, z) = (1.83, 0)$ and $(x, z) = (1.2446, 0.635)$ relative to the centre of initial shoreline.

VOF scheme CICSAM, and detailed implementation of the Cartesian cut-cell method has been discussed for fixed and moving bodies.

In order to validate this method, several benchmark cases with different features have been studied. A 3D collapse of a water column has been computed first. The position of the water front and height of the remaining water surface have been compared with the experimental data and good agreement between numerical and experimental data has been obtained. Thereafter, 3D dambreak flow over a vertical square cylinder has been calculated, where quantitative comparisons for the velocity and force on the cylinder have been made between numerical simulations and corresponding experimental data. The occurence of plunging jet impact and subsequent splash-up have been captured well in the numerical simulation. Then a moving circular cylinder in a quiescent fluid has been studied with the aim to validate the moving body represented by the cut-cell method. Next, a 2D dambreak flow over a wet bed with a moving gate and the water exit and entry of a cylinder have been carried out. The air-water interface profiles have been compared with experimental data and other numerical results, in which good agreement has been obtained. Finally, a 3D water wave generation due to a submerged sliding wedge has been simulated to show the capability of the present 3D two-phase flow model with moving bodies. Snapshots of the air-water interface have been presented and computed water surface profiles showed convincing agreement with experimental data.

This study demonstrates the capability of the present Cartesian cut-cell/volume-of-fluid model to predict 2D and 3D two-phase flow with moving bodies. The model can act as a complementary approach to experimental investigations to gain further insight into the kinematics and dynamics of three-dimensional wave-structure interaction problems. Future research will focus on the fully coupled fluid-structure interaction and also the adaptive mesh method in order to reduce the computational effort.

6. Acknowledgements

We would like to thank the EPSRC projects (number EP/G014264/1 and EP/G015341/1) for helping to fund this work. Z.X. is partialy supported by the the Royal Society Advanced Newton Fellowship (NAF/R1/201156). Constructive comments from anonymous reviewers for the improvement of the manuscript are gratefully acknowledged.

References

[1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567–603.

[2] R. McSherry, K. V. Chua, T. Stoesser, Large eddy simulation of free-surface flows, Journal of Hydrodynamics, Ser. B 29 (2017) 1–12.

[3] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics and Fluids 8 (1965) 2182–2189.

[4] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.

[5] W. J. Rider, B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112–152.

[6] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1992) 25–37.

[7] S. J. Osher, J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1988) 12–49.

[8] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual Review of Fluid Mechanics 35 (2003) 341–372.

[9] D. M. Anderson, G. B. McFadden, A. A. Wheeler, Diffuse-interface methods in fluid mechanics, Annual Review of Fluid Mechanics 30 (1998) 139–165.

[10] Z. Cheng, J. Li, C. Loh, L. Luo, An exactly force-balanced boundary-conforming arbitrary-lagrangian-eulerian method for interfacial dynamics, Journal of Computational Physics 408 (2020) 109–237.

[11] J. J. Monaghan, Smoothed particle hydrodynamics, Annual Review of Astronomy and Astrophysics 30 (1992) 543–574.

[12] J. Li, An arbitrary Lagrangian Eulerian method for three-phase flows with triple junction points, Journal of Computational Physics 251 (2013) 1–16.

[13] Xie, Z., Pavlidis, D., Salinas, P., Pain, C.C., Matar, O.K., A control volume finite element method for three-dimensional three-phase flows, International Journal for Numerical Methods in Fluids (2020). 10.1002/fld.4805.

[14] M. Rudman, Volume-tracking methods for interfacial flow calculations, International Journal for Numerical Methods in Fluids 24 (1997) 671–691.

[15] O. Ubbink, R. I. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, Journal of Computational Physics 153 (1999) 26–50.

[16] Z. Xie, D. Pavlidis, P. Salinas, J. R. Percival, C. C. Pain, O. K. Matar, A balanced-force control volume finite element method for interfacial flows

with surface tension using adaptive anisotropic unstructured meshes, Computers & Fluids 138 (2016) 38–50.

[17] B. P. Leonard, The ultimate conservative difference scheme applied to unsteady one-dimensional advection, Computer Methods in Applied Mechanics and Engineering 88 (1991) 17–74.

[18] J. H. Ferziger, M. Peric, Computational methods for fluid dynamics, 3rd, rev. ed., Springer, Berlin, 2002.

[19] W. Rodi, G. Constantinescu, T. Stoesser, Large-Eddy Simulation in Hydraulics, CRC Press, Boca Raton, Florida, U. S., 2013.

[20] T. Stoesser, Large-eddy simulation in hydraulics: Quo vadis?, Journal of Hydraulic Research 52 (2014) 441–452.

[21] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239–261.

[22] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (2000) 35–60.

[23] Y. H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, Journal of Computational Physics 192 (2003) 593–623.

[24] E. Balaras, Modeling complex boundaries using an external force field

on fixed cartesian grids in large-eddy simulations, Computers & Fluids 33 (2004) 375–404.

[25] M. Kara, T. Stoesser, R. McSherry, Calculation of fluid-structure interaction: methods, refinements, applications, Proceedings of the Institution of Civil Engineers. Engineering and Computational Mechanics 168 (2015) 59–78.

[26] H. S. Udaykumar, W. Shyy, M. M. Rao, Elafint: A mixed eulerian-lagrangian method for fluid flows with complex and moving boundaries, International Journal for Numerical Methods in Fluids 22 (1996) 691–712.

[27] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of Computational Physics 156 (1999) 209–240.

[28] P. G. Tucker, Z. Pan, A cartesian cut cell method for incompressible viscous flow, Applied Mathematical Modelling 24 (2000) 591–606.

[29] D. M. Ingram, D. M. Causon, C. G. Mingham, Developments in cartesian cut cell methods, Mathematics and Computers in Simulation 61 (2003) 561–572.

[30] M. P. Kirkpatrick, S. W. Armfield, J. H. Kent, A representation of curved boundaries for the solution of the navier-stokes equations on a staggered three-dimensional cartesian grid, Journal of Computational Physics 184 (2003) 1–36.

[31] M. H. Chung, Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape, Computers & Fluids 35 (2006) 607–623.

[32] P. Lin, A fixed-grid model for simulation of a moving body in free surface flows, Computers & Fluids 36 (2007) 549–561.

[33] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave-body interactions, Journal of Computational Physics 228 (2009) 6590–6616.

[34] H. Johansen, P. Colella, A cartesian grid embedded boundary method for poisson's equation on irregular domains, Journal of Computational Physics 147 (1998) 60–85.

[35] D. Calhoun, R. J. LeVeque, A cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries, Journal of Computational Physics 157 (2000) 143–180.

[36] M. Popescu, R. Vedder, W. Shyy, A finite volume-based high-order, cartesian cut-cell method for wave propagation, International Journal for Numerical Methods in Fluids 56 (2008) 1787–1818.

[37] L. Duan, X. W. Wang, X. L. Zhong, A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness, Journal of Computational Physics 229 (2010) 7207–7237.

[38] Q. H. Liang, J. Zang, A. G. L. Borthwick, P. H. Taylor, Shallow flow

simulation on dynamically adaptive cut cell quadtree grids, International Journal for Numerical Methods in Fluids 53 (2007) 1777–1799.

[39] K. Fidkowski, D. Darmofal, A triangular cut-cell adaptive method for high-order discretizations of the compressible navierstokes equations, Journal of Computational Physics 225 (2007) 1653–1672.

[40] H. Ji, F. S. Lien, E. Yee, Numerical simulation of detonation using an adaptive cartesian cut-cell method combined with a cell-merging technique, Computers & Fluids 39 (2010) 1041–1057.

[41] D. Hartmann, M. Meinke, W. Schroder, A strictly conservative cartesian cut-cell method for compressible viscous flows on adaptive grids, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1038–1052.

[42] B. Muralidharan, S. Menon, A high-order adaptive cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies, Journal of Computational Physics 321 (2016) 342–368.

[43] M. Meyer, A. Devesa, S. Hickel, X. Y. Hu, N. A. Adams, A conservative immersed interface method for large-eddy simulation of incompressible flows, Journal of Computational Physics 229 (2010) 6300–6317.

[44] Xie, Z., Lin, B., Falconer, R.A., Large-eddy simulation of the turbulent structure in compound open-channel flows, Advances in Water Resources 53 (2013) 66–75.

[45] Xie, Z., Lin, B., Falconer, R.A., Maddux, T.B., Large-eddy simulation

of turbulent open-channel flow over three-dimensional dunes, Journal of Hydraulic Research 51 (2013) 494–505.

[46] F. Gao, D. M. Ingram, D. M. Causon, C. G. Mingham, The development of a cartesian cut cell method for incompressible viscous flows, International Journal for Numerical Methods in Fluids 54 (2007) 1033–1053.

[47] W. Bai, C. G. Mingham, D. M. Causon, L. Qian, Finite volume simulation of viscous free surface waves using the cartesian cut cell approach, International Journal for Numerical Methods in Fluids 63 (2010) 69–95.

[48] T. Q. Li, P. Troch, J. De Rouck, Wave overtopping over a sea dike, Journal of Computational Physics 198 (2004) 686–726.

[49] Z. Xie, Two-phase flow modelling of spilling and plunging breaking waves, Applied Mathematical Modelling 37 (2013) 3698–3713.

[50] Z. Xie, Numerical modelling of wind effects on breaking solitary waves, European Journal of Mechanics - B/Fluids 43 (2014) 135–147.

[51] S. T. Dang, E. A. Meese, J. C. Morud, S. T. Johansen, Numerical approach for generic three-phase flow based on cut-cell and ghost fluid methods, International Journal for Numerical Methods in Fluids 91 (2019) 419–447.

[52] O. Desjardins, J. O. McCaslin, M. Owkes, , P. Brady, Direct numerical and large-eddy simulation of primary atomization in complex geometries, Atomization and Sprays 23 (2013) 1001–1048.

[53] Xie, Z., Lin, B., Falconer, R.A., Turbulence characteristics in free-surface flow over two-dimensional dunes, Journal of Hydro-environmental Research 8 (2014) 200–209.

[54] P. McCorquodale, P. Colella, H. Johansen, A cartesian grid embedded boundary method for the heat equation on irregular domains, Journal of Computational Physics 173 (2001) 620–635.

[55] D. M. Causon, D. M. Ingram, C. G. Mingham, A cartesian cut cell method for shallow water flows with moving boundaries, Advances in Water Resources 24 (2001) 899–911.

[56] G. Dechriste, L. Mieussens, A cartesian cut cell method for rarefied flow simulations around moving obstacles, Journal of Computational Physics 314 (2016) 465–488.

[57] W. P. Bennett, N. Nikiforakis, R. Klein, A moving boundary flux stabilization method for cartesian cut-cell grids using directional operator splitting, Journal of Computational Physics 368 (2018) 333–358.

[58] M. Meinke, L. Schneiders, C. Gunther, W. Schroder, A cut-cell method for sharp moving boundaries in cartesian grids, Computers & Fluids 85 (2013) 135–142.

[59] L. Schneiders, D. Hartmann, M. Meinke, W. Schroder, An accurate moving boundary formulation in cut-cell methods, Journal of Computational Physics 235 (2013) 786–809.

[60] B. Muralidharan, S. Menon, Simulation of moving boundaries interacting with compressible reacting flows using a second-order adaptive

cartesian cut-cell method, Journal of Computational Physics 357 (2018) 230–262.

[61] H. S. Udaykumar, H. C. Kan, W. Shyy, R. TranSonTay, Multiphase dynamics in arbitrary geometries on fixed cartesian grids, Journal of Computational Physics 137 (1997) 366–405.

[62] M. H. Chung, An adaptive cartesian cut-cell/level-set method to simulate incompressible two-phase flows with embedded moving solid boundaries, Computers & Fluids 71 (2013) 469–486.

[63] L. Qian, D. M. Causon, C. G. Mingham, D. M. Ingram, A free-surface capturing method for two fluid flows with moving bodies, Proceedings of the Royal Society a-Mathematical Physical and Engineering Sciences 462 (2006) 21–42.

[64] W. Benguigui, A. Doradoux, J. Lavieville, S. Mimouni, E. Longatte, A discrete forcing method dedicated to moving bodies in twophase flow, International Journal for Numerical Methods in Fluids 88 (2018) 315–333.

[65] Z. Z. Hu, D. M. Causon, C. G. Mingham, L. Qian, A cartesian cut cell free surface capturing method for 3d water impact problems, International Journal for Numerical Methods in Fluids 71 (2013) 1238–1259.

[66] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, B. Buchner, A volume-of-fluid based simulation method for wave impact problems, Journal of Computational Physics 206 (2005) 363–393.

[67] Q. Chen, J. Zang, A. S. Dimakopoulos, D. M. Kelly, C. J. K. Williams, A cartesian cut cell based two-way strong fluid-solid coupling algorithm for 2d floating bodies, Journal of Fluids and Structures 62 (2016) 252–271.

[68] L. Schneiders, C. Gunther, M. Meinke, W. Schroder, An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows, Journal of Computational Physics 311 (2016) 62–86.

[69] P. Heinrich, Nonlinear water waves generated by submarine and aerial landslides, Journal of Waterway, Port, Coastal, and Ocean Engineering-ASCE 118 (1992) 249–266.

[70] T. R. Wu, A numerical study of three-dimensional breaking waves and turbulence effects, Ph.D. thesis, Cornell University, 2004.

[71] J. Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment, Monthly Weather Review 91 (1963) 99–164.

[72] D. K. Lilly, A proposed modification of the germano-subgrid-scale closure method, Physics of Fluids A 4 (1992) 633–635.

[73] C. Hirsch, Numerical computation of internal and external flows introduction to the fundamentals of CFD, new ed., Butterworth-Heinemann, Oxford, 2007.

[74] Z. Xie, Numerical study of breaking waves by a two-phase flow model, International Journal for Numerical Methods in Fluids 70 (2012) 246–268.

[75] H. A. van der Vorst, Bi-CGSTAB - a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear-systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631–644.

[76] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1986) 40–65.

[77] Z. Xie, Numerical modelling of breaking waves under the influence of wind, Ph.D. thesis, University of Leeds, 2010.

[78] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112–152.

[79] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567–603.

[80] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. thesis, Imperial College of Science, Technology and Medicine, 1997.

[81] D. M. Causon, D. M. Ingram, C. G. Mingham, G. Yang, R. V. Pearson, Calculation of shallow water flows using a cartesian cut cell approach, Advances in Water Resources 23 (2000) 545–562.

[82] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, Computer Graphics 21 (1987) 163–169.

[83] P. L. Roe, Characteristic-based schemes for the Euler equations, Annual Review of Fluid Mechanics 18 (1986) 337–365.

[84] N. P. Waterson, H. Deconinck, Design principles for bounded higher-order convection schemes - a unified approach, Journal of Computational Physics 224 (2007) 182–207.

[85] S. V. Patankar, Numerical heat transfer and fluid flow, Taylor & Francis,, London, 1980.

[86] J. C. Martin, W. J. Moyce, An experimental study of the collapse of liquid columns on a rigid horizontal plane .4., Philosophical Transactions of the Royal Society of London Series a-Mathematical and Physical Sciences 244 (1952) 312–324.

[87] D. Pavlidis, J. L. M. A. Gomes, Z. Xie, J. R. Percival, C. C. Pain, O. K. Matar, Compressive advection and multi-component methods for interface-capturing, International Journal of Numerical Methods in Fluids 80 (2016) 256–282.

[88] D. Tritton, Experiments on the flow past a circular cylinder at low reynolds number, Journal of Fluid Mechanics 6 (1959) 547–567.

[89] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, Journal of Fluid Mechanics 98 (1980) 819–855.

[90] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface cartesian grid method i: An easily implemented technique for 3d moving boundary computations, Journal of Computational Physics 210 (2005) 1–31.

[91] A. A. Shirgaonkar, M. A. MacIver, N. A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, Journal of Computational Physics 228 (2009) 2366–2390.

[92] J. Wu, C. Shu, Y. H. Zhang, Simulation of incompressible viscous flows around moving objects by a variant of immersed boundarylattice boltzmann method, International Journal of Numerical Methods in Fluids 62 (2010) 327–354.

[93] I. M. Janosi, D. Jan, K. G. Szabo, T. Tel, Turbulent drag reduction in dam-break flows, Experiments in Fluids 37 (2004) 219–229.

[94] Greenhow, M., Moyo, S., Water entry and exit of horizontal circular cylinders, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 355 (1997) 551–563.

[95] P. L. F. Liu, T. R. Wu, F. Raichlen, C. E. Synolakis, J. C. Borrero, Runup and rundown generated by three-dimensional sliding masses, Journal of Fluid Mechanics 536 (2005) 107–144.

[96] Z. Xie, A two-phase flow model for three-dimensional breaking waves over complex topography, Proceedings of the Royal Society A: Mathematical, Physical & Engineering Sciences 471 (2015) 20150101.