Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/131503/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Xie, Zhihua ORCID: <https://orcid.org/0000-0002-5180-8427> and Stoesser, Thorsten ORCID: <https://orcid.org/0000-0001-8874-9793> 2020. A three-dimensional Cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies. Journal of Computational Physics 416 , 109536. 10.1016/j.jcp.2020.109536 file

Publishers page: <http://dx.doi.org/10.1016/j.jcp.2020.109536>
<<http://dx.doi.org/10.1016/j.jcp.2020.109536>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

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 exit 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 effi-

21 ciency and easy implementation. From a general point of view, there are two
 22 classes of algorithms to solve the transport equation of the volume fraction:
 23 geometric and algebraic computation [5]. In the geometric VOF methods
 24 [1], interfaces are first reconstructed from the volume fraction data so that
 25 a geometric profile is found which approximates the actual interface loca-
 26 tion. Then changes in volume fraction are calculated by integrating volume
 27 fluxes across cell boundaries, using flux splitting or unsplitting schemes. In
 28 the algebraic computation [14–16], the interface is captured by solving the
 29 transport equation of the volume fraction with a differencing scheme with-
 30 out reconstructing the interface, such as the flux-corrected transport scheme
 31 [14] and using the normalised variable diagram (NVD) [17] concept to switch
 32 between different differencing schemes [15].

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

46 Due to the high demand of DNS in terms of the number of grid points (which
 47 is proportional to $Re^{9/4}$ in 3D, where Re is the Reynolds number), it has been
 48 mostly used for relatively low Reynolds number flows or flows in a relatively
 49 small flow domain. As DNS is out of reach for practical applications, recent
 50 developments have focused on the method of large-eddy simulations (LES)
 51 [19, 20], in which the large-scale eddies are resolved, i.e. calculated directly,
 52 while the effects of the small eddies on the large-scale turbulence are approx-
 53 imated. LES has been already employed for practical problems where the
 54 Reynolds number is high and the computational domain is large.

55 To deal with complex geometries with fixed or moving motion in engineer-
 56 ing applications, overlapping grids, boundary-fitted grids and unstructured
 57 grids can be used. Unstructured grids provide great flexibility in conform-
 58 ing to complex boundaries, and can easily be refined or coarsened in specific
 59 regions of the flow domain depending on the flow feature. However, they re-
 60 quire additional computational efforts and further complicate the algorithm
 61 implementation as there is no pre-defined order of the control volumes and
 62 their geometric layouts need to be calculated. Furthermore, generating high-
 63 quality boundary-fitted or unstructured grids is usually very cumbersome
 64 [21], especially for moving body problems, where the mesh has to be regen-
 65 erated at every time step. Cartesian grid methods which can simulate flow
 66 with complex geometries on fixed Cartesian grids, avoid these problems. The
 67 primary advantage of the Cartesian grid method is that only little modifica-
 68 tion of the flow solver on Cartesian grids is needed to account for complex
 69 (immersed) geometries. It also has the advantage of simplified grid genera-
 70 tion and simulating flows with moving boundaries whilst avoiding deforming

71 grids. The two most popular methods are the immersed boundary method
72 [21–25] and the Cartesian cut-cell method [26–31]. There are some examples
73 for two-phase flows with moving bodies using immersed boundary method
74 in two-dimensional (2D) [32] and three-dimensional (3D) [33] simulations.
75 Compared to immersed boundary methods, the Cartesian cut-cell method
76 is very attractive as it enforces strict conservation of mass, momentum and
77 energy, and in particular near solid boundaries. The present study focuses on
78 the Cartesian cut-cell method hence only relevant studies using this method
79 are mentioned hereafter.

80 The Cartesian cut-cell method is based on a stationary Cartesian back-
81 ground grid, in which the solid boundary is intersected with boundary cells
82 (named as cut-cells), and regular grid cells are truncated to conform to the
83 solid (immersed) boundary interface. There is no modification for standard
84 regular grid cells whereas special treatments are needed for the cut-cells. For
85 fixed, solid boundaries, the Cartesian cut-cell method has been developed
86 for 2D Poisson [34], and advection-diffusion [35] equations. It has also been
87 applied to study single-phase fluid flow problems, such as for 2D aeroacous-
88 tics [36], 2D hypersonic boundary layer transition [37], 2D shallow water
89 equations [38], 2D incompressible viscous flow [27, 28, 31], 2D compress-
90 ible viscous flow [39], 3D inviscid flow [40], 3D incompressible viscous flow
91 [30], 3D compressible viscous flow [41, 42], and 3D LES studies [43–45]. The
92 Cartesian cut-cell method has also been used to study free-surface/two-phase
93 flows in 2D using a two-fluid approach [46], height function [47], volume-of-
94 fluid method [48–50], in the form of a coupled level-set and volume-of-fluid
95 approach [51], and 3D DNS and LES studies [52, 53].

96 For moving bodies, Cartesian cut-cell methods have been developed for
 97 solving the 2D heat [54] and 2D shallow water equations [55], 2D viscous flow
 98 [26], 3D rarefied gas flows [56], 3D inviscid flow [57], and 3D compressible
 99 viscous flow [58–60]. Some early development of the cut-cell method for
 100 single-phase flows can be found in [29]. For 2D two-phase flows, different
 101 approaches have been employed for the interface tracking/capturing, such
 102 as the Lagrangian marker [61], level-set method [62], and the two-fluid
 103 approach [63, 64]. In 3D, the Euler equations have been solved together
 104 with a density function to capture the air-water interface for a water impact
 105 problem in [65] and a cut-cell method with moving body has been developed
 106 to study the 3D wave impact problem with a single-phase volume-of-fluid
 107 method in [66]. Recently, there are some development for fluid-structure
 108 interaction problems for 2D two-phase flow [67] and 3D single-phase flow
 109 [68]. To date, there has been relatively little work on cut-cell methods with
 110 moving bodies for the 3D Navier–Stokes equations with two-phase flows and
 111 turbulence.

112 The objective of this paper is, therefore, to present and validate thor-
 113 oughly an efficient finite volume method based on the Cartesian cut-cell
 114 method for the unsteady, turbulent, incompressible, two-phase Navier–Stokes
 115 equations with moving bodies on a three-dimensional, non-uniform, stag-
 116 gered, Cartesian grid. The method of large-eddy simulation is employed to
 117 compute directly large-scale turbulence of the flow by solving the filtered
 118 Navier–Stokes equations and employing the dynamic Smagorinsky sub-grid
 119 scale model to account for the unresolved (subgrid) scales of turbulence.
 120 An algebraic VOF scheme is employed and modified in cut-cells to capture

121 the air-water interface in the two-phase flow model. Solid boundaries are re-
122 solved by the Cartesian cut-cell method, with detailed 3D cut-cell generation
123 and finite volume discretisation. The moving body algorithm (as an internal
124 source function by taking the solid volume change into account) is imple-
125 mented to simulate moving bodies on a fixed Cartesian grid. Moreover, an
126 implicit time integration scheme is used for time integration, which prevents
127 common instability problems in small cut-cells as reported in the literature.

128 The organisation of this paper is as follows. The description of the math-
129 ematical model for the two-phase flow is described in [Section 2](#). The nu-
130 merical method and implementation of the cut-cell method are presented
131 in [Section 3](#). The versatility, robustness and accuracy of the present two-
132 phase flow model is demonstrated by solving various 2D and 3D two-phase
133 flow problems with moving bodies in [Section 4](#). Finally, the paper ends with
134 conclusions in [Section 5](#).

135 **2. Mathematical model**

136 *2.1. Governing equations*

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

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

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

139 where $\mathbf{u}$ is the velocity vector, t is the time, p is the pressure, $\mathbf{g}$ is the grav-
140 itational acceleration vector, ρ and μ are the density and dynamic viscosity

141 of the fluid.

142 2.2. Interface modelling

143 2.2.1. Volume-of-fluid method

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

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

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

$$\frac{dF}{dt} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0. \quad (4)$$

150 2.2.2. Physical properties

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

$$\rho = F\rho^w + (1 - F)\rho^a, \quad (5)$$

$$\mu = F\mu^w + (1 - F)\mu^a, \quad (6)$$

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

155 2.3. Moving body algorithm

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

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

where $\psi(t)$ is an 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:

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

165 2.4. Subgrid-scale model

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

$$\nabla \cdot \bar{\mathbf{u}} = \psi(t), \quad (9)$$

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.

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

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

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

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

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

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

179 with the cut-off length scale $\bar{\Delta} = (\Delta x \Delta y \Delta z)^{1/3}$ and the coefficient $C_d =$
 180 $\frac{1}{2} \frac{L_{ij} M_{ij}}{M_{ij} M_{ij}}$ is calculated by the dynamic Smagorinsky model [72] in the present
 181 study, where $L_{ij} = \widehat{\widehat{u_i u_j}} - \widehat{\widehat{u_i}} \widehat{\widehat{u_j}}$ and $M_{ij} = \widehat{\widehat{\Delta}^2 |\widehat{\widehat{\mathbf{S}}}| \widehat{\widehat{S_{ij}}} - \bar{\Delta}^2 |\widehat{\widehat{\mathbf{S}}}| \widehat{\widehat{S_{ij}}}}$, and the
 182 hat $\widehat{\cdot}$ represents spatial filtering over the test filter. The symbol for spa-
 183 tial filtering ‘ $\widehat{\cdot}$ ’ is dropped hereinafter for simplicity. The advantage of the
 184 dynamic Smagorinsky model is that the dissipation of energy from the large-
 185 scale turbulence is approximated in analogy to dissipation on a molecular
 186 level without the need for empirical input (due to the dynamic approxima-
 187 tion of the Smagorinsky constant) and hence has found to work well for many
 188 applications of practical interest [19].

189 2.5. Initial and boundary conditions

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

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

204 3. Numerical method

205 3.1. Finite volume discretisation

206 One option of discretising the governing equations is the finite volume
207 method (FVM). In the FVM, also known as the control volume method, the
208 entire flow domain is divided into a number of control volumes surrounding
209 each grid point. The differential equation is integrated over each control vol-
210 ume (CV) in order to derive the algebraic equation containing the grid-point
211 values of φ , where φ is the dependent variable. The FVM is conservative
212 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 $\mathbf{n}$. All the governing equations can be recast into a general integral formulation as below

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

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	$\mathbf{u}$	$\mu + \mu_t$	$-\nabla p + \rho\mathbf{g} + \rho\mathbf{u}\psi(t)$
Volume fraction	F	0	$\rho F \nabla \cdot \mathbf{u}$

226 3.2. Computational grid

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

Thus, the CV's volume is obtained as

$$\Omega = \iiint_{\Omega} d\Omega = \Delta x \Delta y \Delta z. \quad (14)$$

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

$$A_e = \iint_e dS = \Delta y \Delta z. \quad (15)$$

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

$$\varphi_e = \lambda_e \varphi_P + (1 - \lambda_e) \varphi_E, \quad (16)$$

where λ_e is the interpolation factor defined as

$$\lambda_e = \frac{|eE|}{|PE|}. \quad (17)$$

238 Analogous expressions can be derived for all other faces (f=w, n, s, b, r)
 239 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 $\phi(i, j, k)$ are stored at the centre of the CV. Δx , Δy , and Δz are the CV's length, and $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ 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

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

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](#).

248 3.3.2. Spatial discretisation

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

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

$$a_P^\varphi \varphi_P^{n+1} = \sum a_{nb}^\varphi \varphi_{nb}^{n+1} + b_P^\varphi, \quad (19)$$

258 where a^φ is the coefficient, the subscripts P and nb = E, W, N, S, B, R denote
 259 the variables in the present and neighbouring cells respectively and b_P^φ is the
 260 source term.

261 The algebraic equations are solved by the SIP (Strongly Implicit Proce-
 262 dure) method or Bi-CGSTAB (Bi-Conjugate Gradients Stablized) Method
 263 [75] in this study.

264 3.3.3. Pressure-velocity coupling

265 In the incompressible Navier–Stokes equations pressure and velocity are
 266 decoupled as the pressure term does not appear in the continuity equation.
 267 For some numerical discretisations this may cause convergence problems.
 268 However, when a staggered mesh is used, as in this work, coupling occurs
 269 as a result of the discretisation, as velocity updates on cell faces contain

270 pressure terms. In this study, the PISO algorithm [76] is employed for the
 271 pressure-velocity coupling. The PISO algorithm is used to calculate the
 272 corrected pressure twice and after solving the pressure correction equations,
 273 the updated pressure and velocity are added by the pressure and velocity
 274 correction terms respectively. Here only a brief summary is presented and
 275 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 $\mathbf{u}^*$, which satisfy

$$a_P^u \mathbf{u}_f^* = \sum a_{nb}^u \mathbf{u}_{nb}^* + b_P^u + A_f(p_P^* - p_{nb}^*). \quad (20)$$

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

$$a_P^p p'_P = \sum a_{nb}^p p'_{nb} + b'_P, \quad (21)$$

276 where the term b'_P , called the mass residual, in the pressure correction equation
 277 is the left-hand side of the discretised continuity equation evaluated in
 278 terms of the fluid velocities $\mathbf{u}^*$.

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

$$a_P^p p''_P = \sum a_{nb}^p p''_{nb} + b''_P, \quad (22)$$

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

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

updated as

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

where

$$\begin{aligned} \mathbf{u}'_f &= \frac{A_f}{a_P^u} (p'_P - p'_{nb}), \\ \mathbf{u}''_f &= \frac{\sum a_{nb}^u \mathbf{u}'_{nb} + A_f (p''_P - p''_{nb})}{a_P^u}. \end{aligned} \tag{24}$$

282 3.4. VOF solver for interface capturing

283 A key requirement for simulating two-phase flows is a method for calcu-
 284 lating the shape of the interface. Numerous methods have been proposed and
 285 used for the simulation of free-surface/two-phase flows. However, the VOF
 286 method for capturing the interface is one of the most popular approaches
 287 due to its advantages: mass conservation, computational efficiency and easy
 288 implementation. From a general point of view, there are two classes of algo-
 289 rithms to solve the F transport equation (Eq. (4)): algebraic and geometric
 290 computation [78]. Excellent reviews on the VOF methods can be found in
 291 [78, 79].

292 Considering the advantages of the VOF method and efficiency in alge-
 293 braic computation, the high resolution VOF scheme Compressive Interface
 294 Capturing Scheme for Arbitrary Meshes (CICSAM) is employed in this study
 295 to capture the air-water interface for two-phase flows. CICSAM is a high res-
 296 olution scheme based on the normalised variable diagram used by Leonard
 297 [17]. It contains two high resolution schemes and the weighting factor is
 298 based on the angle between the interface and the direction of motion. An
 299 outline of CICSAM is given below. Refer to [80] for more details.

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

$$\tilde{F} = \frac{F - F_U}{F_A - F_U}, \quad (25)$$

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

$$\tilde{F}_{f_{CBC}} = \begin{cases} \min \left\{ 1, \frac{\tilde{F}_D}{c_D} \right\}, & \text{when } 0 \leq \tilde{F}_D \leq 1 \\ \tilde{F}_D, & \text{when } \tilde{F}_D < 0, \tilde{F}_D > 1 \end{cases} \quad (26)$$

305 where subscript D indicates donor cell, $c_D = \sum_f \max \left\{ \frac{-V_f \Delta t}{\Omega_D}, 0 \right\}$ is the Courant
 306 number of the donor cell and V_f is the volumetric flux. However, the Hyper-C
 307 scheme is inadequate to preserve the shape of an interface which lies tangen-
 308 tially to the flow direction. Thus CICSAM switches to the ULTIMATE-
 309 QUICKEST (UQ) scheme [17]

$$\tilde{F}_{f_{UQ}} = \begin{cases} \min \left\{ \frac{8c_D \tilde{F}_D + (1 - c_D)(6\tilde{F}_D + 3)}{8}, \tilde{F}_{f_{CBC}} \right\}, & \text{when } 0 \leq \tilde{F}_D \leq 1 \\ \tilde{F}_D, & \text{when } \tilde{F}_D < 0, \tilde{F}_D > 1 \end{cases} \quad (27)$$

310 in this case.

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

312 SAM combines these two schemes, then

$$\tilde{F}_f = \gamma_f \tilde{F}_{f_{\text{CBC}}} + (1 - \gamma_f) \tilde{F}_{f_{\text{UQ}}}, \quad (28)$$

313 in which the weighting factor is given as

$$\gamma_f = \min \left\{ k_\gamma \frac{\cos(2\alpha_\gamma) + 1}{2}, 1 \right\}, \quad (29)$$

314 where k_γ is a constant introduced to control the dominance of the different
 315 schemes and the recommended value is $k_\gamma = 1$, α_γ is the angle between the
 316 vector normal to the interface and the vector which convects the centres of
 317 donor and acceptor cells.

318 The final expression for the face value of F is

$$F_f = (1 - \beta_f) F_D + \beta_f F_A, \quad (30)$$

319 where the weight factor β_f is obtained by

$$\beta_f = \frac{\tilde{F}_f - \tilde{F}_D}{1 - \tilde{F}_D}. \quad (31)$$

320 It is noted that the normalised variable in [Eq. \(25\)](#) will be divided by
 321 zero if the volume fraction F has the same value in the acceptor and upwind
 322 cell. In the numerical implementation, the numerator and denominator of
 323 the weighting factor in [Eq. \(31\)](#) are multiplied by $(F_A - F_U)$, resulting in a
 324 modified expression for the normalised variable on the face (not shown here),
 325 in order to avoid the singularity in the computation. In the present study,
 326 for cut-cells, the flux across each face of the control volume is modified when
 327 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:

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

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

350 In order to calculate the area and volume of the cut-cell, how the solid
351 boundary cuts the Cartesian cell needs to be determined first. The marching
352 tube algorithm [82] is employed and the grid nodes can be classified as inside
353 or outside of the solid based on the sign of the level set function $LS(x, y, z, t)$.
354 For 3D, there are 8 grid nodes in each cell so that 2^8 configurations are
355 possible. By using the inversion and rotation of different configurations, the
356 256 cases can be reduced to 15 cases as shown in Fig. 3 for a typical cut cell
357 in a 3D Cartesian grid, which includes zero, one, two, three, and four points
358 being cut through the Cartesian grid cell.

359 Once the configuration of the cut cell has been determined, the intersec-
360 tion point along each edge can be obtained by linear interpolation of the level
361 set function of two neighbouring grid nodes. Once the point of intersection
362 of the line with the cut plane is found, the geometric information can be
363 calculated. Finally, the face areas and total volume of the truncated cell are
364 calculated by numerical integration. In contrast to a full fluid cell, the spatial
365 discretisation at cell faces and cell centres are modified in a cut cell, which
366 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.

367 *3.5.2. Advection term*

368 The finite volume discretisation of the advection term in [Eq. \(13\)](#) is ob-
 369 tained as

$$\begin{aligned}
 Q_\varphi^C &= \iint_S (\rho \mathbf{u} \cdot \mathbf{n}) \varphi dS = \sum_f (\theta A)_f [(\rho \mathbf{u} \cdot \mathbf{n}) \varphi]_f \\
 &= \sum_f (\rho \mathbf{u} \cdot \mathbf{n} \theta A)_f \varphi_f \\
 &= \sum_f m_f \varphi_f,
 \end{aligned} \tag{33}$$

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

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

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

$$m_f = 0. \tag{35}$$

375 The mass flux at the faces of the momentum control volume can be ob-
 376 tained by the interpolation of values of ρ and $\mathbf{u}$, such as $m_f = \rho_f \mathbf{u}_f \cdot \mathbf{n} \theta_f A_f$,
 377 however, the mass conservation in the momentum control volume can be only
 378 guaranteed to the accuracy of the interpolation procedure [\[18\]](#). Thus, in this
 379 study, the m_f is obtained from the interpolation of the mass fluxes, which is
 380 already available at the faces of the continuity control volumes.

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

$$\varphi_f^{\text{HRS}} = \varphi_f^{\text{FOU}} + \Phi(r_f^m) \Psi(r_f^\varphi) (\varphi_f^{\text{SOU}} - \varphi_f^{\text{FOU}}), \quad (36)$$

386 in which φ_f^{FOU} and φ_f^{SOU} are the corresponding values obtained from the
 387 first-order and second-order upwind schemes.

388 $\Psi()$ is the limiter function where the minmod limiter [83], which is one
 389 of the simplest second-order TVD (Total Variation Diminishing) schemes, is
 390 used here

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

391 Similar results are obtained by using other limiter functions and a general re-
 392 view of various limiter functions can be found in [84] and will not be repeated
 393 here. r_f^φ represents the ratio of successive gradients of φ on the solution mesh
 394 and obtained as

$$r_f^\varphi = \frac{\varphi_A - \varphi_D}{\varphi_D - \varphi_U}, \quad (38)$$

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

397 As discussed in the previous study [74], $\Phi()$ is the step function and r_f^m
 398 is the variation for the mass, defined as the ratio of the mass flux between

399 the conservative and non-conservative forms

$$r_f^m = \frac{(\rho \mathbf{u} \cdot \mathbf{n} \theta A)_f}{\rho_c (\mathbf{u} \cdot \mathbf{n} \theta A)_f}. \quad (39)$$

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

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

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

404 3.5.3. Diffusion term

405 The finite volume discretisation of the diffusion term in [Eq. \(13\)](#) is ob-
 406 tained as

$$Q_\varphi^D = \iint_S \Gamma \frac{\partial \varphi}{\partial n} dS = \sum_f \Gamma_f \frac{\partial \varphi}{\partial n} (\theta A)_f, \quad (41)$$

407 where the viscosity on the face is obtained by the harmonic mean [\[85\]](#), for
 408 example, on the east face

$$\Gamma_e = \frac{\Gamma_P \Gamma_E}{\lambda_e \Gamma_P + (1 - \lambda_e) \Gamma_E}, \quad (42)$$

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

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

$$\frac{\partial \varphi}{\partial n} = \frac{\varphi_{\text{nb}} - \varphi_{\text{P}}}{\Delta_{\text{Pnb}}}, \quad (43)$$

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

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

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

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

420 3.5.4. Source term

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

423 *Pressure term.* The finite volume discretisation of the pressure term is ob-
424 tained as

$$Q_{\varphi}^{\text{p}} = \iiint_{\Omega} -\nabla p d\Omega = -\nabla p \theta_c \Omega, \quad (45)$$

425 and the pressure gradient is calculated as

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

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

$$Q_{\varphi}^B = \iiint_{\Omega} \rho \mathbf{g} d\Omega = \rho_c \mathbf{g} \theta_c \Omega, \quad (47)$$

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

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

$$Q_{\varphi}^I = \iiint_{\Omega} \rho \mathbf{u} \psi(t) d\Omega = \rho_c \mathbf{u} \psi_c(t) \theta_c \Omega. \quad (48)$$

432 3.6. Procedure

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

- 435 (1) Setup boundary conditions and moving bodies with pre-specified mo-
 436 tion: [Section 2.5](#).
- 437 (2) Calculate the cut-cell information and the θ function for the area and
 438 volume available for the flow in each cells, and compute the internal
 439 source function $\psi(t)$: [Section 3.5.1](#).
- 440 (3) Solve the volume-of-fluid function F : [Section 3.4](#).
- 441 (4) Update the fluid properties, density ρ and dynamic viscosity μ : [Section 2.2.2](#).
- 442 (5) Calculate the turbulent eddy viscosity μ_t if the dynamic SGS model is
 443 used: [Section 2.4](#).

- 444 (6) Setup the coefficients a^u and b_p^u based on the latest pressure p^* and
 445 velocity $\mathbf{u}$ and solve for the intermediate velocity $\mathbf{u}^*$: [Section 3.3.3](#).
- 446 (7) Setup the coefficients a^p and b_p' and solve for the first pressure correc-
 447 tion p' and calculate $\mathbf{u}'$: [Section 3.3.3](#).
- 448 (8) Setup the coefficients b_p'' and solve for the second pressure correction
 449 p'' and calculate $\mathbf{u}''$: [Section 3.3.3](#).
- 450 (9) Update the pressure $p = p^* + p' + p''$ and the velocity $\mathbf{u}^* + \mathbf{u}' + \mathbf{u}''$:
 451 [Section 3.3.3](#).
- 452 (10) Go back to step (6) if the mass residue is larger than a threshold,
 453 otherwise go to next time level in step (1): [Section 3.3](#).

454 4. Results and discussion

455 A grid convergence study of the two-phase flow model, the benchmark
 456 problem of a 3D collapse of a water column without a structure is carried out
 457 first to test the interface capturing method at various grid resolutions. The
 458 validation of the cut-cell method is then demonstrated by studying dambreak
 459 flow interacting with a fixed square cylinder. Numerical results are quantita-
 460 tively compared with experimental data in terms of velocity and force acting
 461 on the cylinder. With the goal to validate the cut-cell method involving mov-
 462 ing bodies, a single-phase flow problem for a moving cylinder in a quiescent
 463 fluid is studied followed by a two-phase flow with moving bodies. Therefore,
 464 2D dambreak flow over a wet bed is studied with the movement of the gate
 465 being simulated directly using the proposed moving cut-cell method, and the

466 computed water surface behavior is compared qualitatively with experimen-
 467 tal observations. After that, the validation of the Cartesian cut-cell/VOF
 468 method with moving bodies is further demonstrated by studying the exit
 469 and entry of a circular cylinder into a fluid. Finally, the method is employed
 470 to study 3D water wave generation due to a sliding wedge.

471 *4.1. Convergence study - 3D collapse of a water column*

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

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

485 Fig. 4 depicts snapshots of the remaining water height (left) and surge
 486 front position (right) as computed together with experimental data [86]. As
 487 far as the water column height is concerned, simulation results on all three
 488 grids are in very good agreement with the experimental data (Fig. 4, left).
 489 As far as the surge front position is concerned, due to the fact that the
 490 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.

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

507 4.2. 3D dambreak flow interacting with a fixed square cylinder

508 In this section, a 3D two-phase flow interacting with a fixed structure is
 509 considered to verify and validate the implementation of the Cartesian cut-
 510 cell method. The dambreak flow interacting with a square cylinder is often
 511 considered as a benchmark case and this has been studied in [70], among
 512 others. The computational setup (shown in Fig. 6) is the same as in [70]
 513 except that a full domain is considered in the present study, rather than
 514 using only half of the domain with symmetry boundary conditions along
 515 the centre of the domain as was used by [70]. The tank dimensions are 1.6
 516 m $\times$ 0.61 m $\times$ 0.60 m in the streamwise, spanwise, and vertical direction,
 517 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.

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

526 Fig. 7 presents snapshots of the air-water interface as the flow interacts
 527 with the square cylinder. Initially, the water column collapses due to gravity
 528 and a two-dimensional bore is generated. When the bore hits the cylinder,
 529 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.

568 4.4. Dambreak over a wet bed with a moving gate

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

582 The calculated progression of the dambreak flow over a wet bed is vi-
 583 sualised by the blue contours indicating the water face in the right column
 584 of Fig. 12. Also depicted in the left column are photographs of the experi-

ment 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/s² and the upward velocity of the cylinder is $U_{\text{cylinder}} = 0.39$ m/s. As the focus is on the impact region, a

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.

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

615 Fig. 13 shows snapshots of the air-water interface together with the cylin-
616 der for the water exit problem. As the cylinder reaches $T = 0.2$, the air-water
617 interface deforms in the region above the cylinder. At $T = 0.4$ and $T = 0.6$,
618 more significant water surface deformation is observed taking the shape of
619 the cylinder in its vicinity. At these two time instants, the predicted results
620 compare well with the BEM results by [94] (red symbols). At $T = 0.8$ and
621 $T = 1.0$, a thin layer of fluid is found on the surface of the cylinder and
622 as displaced fluid flows back into the water body a depression of the water
623 surface is observed on either side of the cylinder. The thin layer of fluid
624 around the cylinder eventually dries up ($T = 2.0$) before the cylinder exists
625 the water completely ($T = 3.0$). The present results are similar to those
626 obtained by the single-phase flow model of [32] and very close to those given
627 by the two-phase flow model of [33]. The thin water film over the cylinder
628 appears better resolved with two-phase flow models.

629 For the water entry problem, the overall setup is the same as the water
630 exist problem except that the cylinder starts at a distance $d = 1.25$ m above
631 the still water level and moves with a constant downward velocity $U_{\text{cylinder}} =$
632 -1.0 m/s.

633 Fig. 14 shows profiles of the air-water interface at selected instants in
634 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).

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

642 *4.6. Water wave generation by a 3D sliding wedge*

643 The water waves generated by a 3D sliding wedge has been studied both
644 experimentally and numerically in [95]. In the experiments, the initial po-
645 sition of the wedge varied from subaerial to submerged. The subaerial case
646 has been studied in [33, 95] and here the focus is on the submerged wedge
647 case. The setup is the same as in the experiment and details can be found in
648 [95]. The computational domain of $6.6 \text{ m} \times 3.7 \text{ m} \times 3.3 \text{ m}$ is discretised by
649 a $220 \times 160 \times 220$ uniform Cartesian grid in the streamwise, spanwise, and
650 vertical directions, respectively, with its origin being setup at the shoreline
651 along the central plane. The motion of the sliding wedge in the simulation
652 is prescribed and is the same as in the experiment.

653 Fig. 15 shows snapshot of the air-water interface profile together with
654 the location of the wedge as it moves down the slope. At the beginning, the
655 submerged wedge induces downward velocity above the wedge, which causes
656 a depressed air-water interface above the wedge and push the water surface
657 upward in front of the wedge. A small positive wave and a large negative
658 wave are generated as the wedge reaches its full speed. A positive wave is
659 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 selected 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 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.

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 occurrence 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

697 represented by the cut-cell method. Next, a 2D dambreak flow over a wet
698 bed with a moving gate and the water exit and entry of a cylinder have been
699 carried out. The air-water interface profiles have been compared with exper-
700 imental data and other numerical results, in which good agreement has been
701 obtained. Finally, a 3D water wave generation due to a submerged sliding
702 wedge has been simulated to show the capability of the present 3D two-
703 phase flow model with moving bodies. Snapshots of the air-water interface
704 have been presented and computed water surface profiles showed convincing
705 agreement with experimental data.

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

713 **6. Acknowledgements**

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

719 **References**

- 720 [1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface
721 and interfacial flow, *Annual Review of Fluid Mechanics* 31 (1999) 567–
722 603.
- 723 [2] R. McSherry, K. V. Chua, T. Stoesser, Large eddy simulation of free-
724 surface flows, *Journal of Hydrodynamics, Ser. B* 29 (2017) 1–12.
- 725 [3] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent
726 viscous incompressible flow of fluid with free surface, *Physics and Fluids*
727 8 (1965) 2182–2189.
- 728 [4] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dy-
729 namics of free boundaries, *Journal of Computational Physics* 39 (1981)
730 201–225.
- 731 [5] W. J. Rider, B. Kothe, Reconstructing volume tracking, *Journal of*
732 *Computational Physics* 141 (1998) 112–152.
- 733 [6] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous,
734 incompressible, multi-fluid flows, *Journal of Computational Physics* 100
735 (1992) 25–37.
- 736 [7] S. J. Osher, J. A. Sethian, Fronts propagating with curvature dependent
737 speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of*
738 *Computational Physics* 79 (1988) 12–49.
- 739 [8] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, *Annual*
740 *Review of Fluid Mechanics* 35 (2003) 341–372.

- 741 [9] D. M. Anderson, G. B. McFadden, A. A. Wheeler, Diffuse-interface
742 methods in fluid mechanics, *Annual Review of Fluid Mechanics* 30
743 (1998) 139–165.
- 744 [10] Z. Cheng, J. Li, C. Loh, L. Luo, An exactly force-balanced boundary-
745 conforming arbitrary-lagrangian-eulerian method for interfacial dynam-
746 ics, *Journal of Computational Physics* 408 (2020) 109–237.
- 747 [11] J. J. Monaghan, Smoothed particle hydrodynamics, *Annual Review of*
748 *Astronomy and Astrophysics* 30 (1992) 543–574.
- 749 [12] J. Li, An arbitrary Lagrangian Eulerian method for three-phase flows
750 with triple junction points, *Journal of Computational Physics* 251 (2013)
751 1–16.
- 752 [13] Xie, Z., Pavlidis, D., Salinas, P., Pain, C.C., Matar, O.K., A con-
753 trol volume finite element method for three-dimensional three-phase
754 flows, *International Journal for Numerical Methods in Fluids* (2020).
755 [10.1002/flid.4805](https://doi.org/10.1002/flid.4805).
- 756 [14] M. Rudman, Volume-tracking methods for interfacial flow calculations,
757 *International Journal for Numerical Methods in Fluids* 24 (1997) 671–
758 691.
- 759 [15] O. Ubbink, R. I. Issa, A method for capturing sharp fluid interfaces on
760 arbitrary meshes, *Journal of Computational Physics* 153 (1999) 26–50.
- 761 [16] Z. Xie, D. Pavlidis, P. Salinas, J. R. Percival, C. C. Pain, O. K. Matar, A
762 balanced-force control volume finite element method for interfacial flows

- 763 with surface tension using adaptive anisotropic unstructured meshes,
764 Computers & Fluids 138 (2016) 38–50.
- 765 [17] B. P. Leonard, The ultimate conservative difference scheme applied to
766 unsteady one-dimensional advection, Computer Methods in Applied
767 Mechanics and Engineering 88 (1991) 17–74.
- 768 [18] J. H. Ferziger, M. Peric, Computational methods for fluid dynamics,
769 3rd, rev. ed., Springer, Berlin, 2002.
- 770 [19] W. Rodi, G. Constantinescu, T. Stoesser, Large-Eddy Simulation in
771 Hydraulics, CRC Press, Boca Raton, Florida, U. S., 2013.
- 772 [20] T. Stoesser, Large-eddy simulation in hydraulics: Quo vadis?, Journal
773 of Hydraulic Research 52 (2014) 441–452.
- 774 [21] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review
775 of Fluid Mechanics 37 (2005) 239–261.
- 776 [22] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined
777 immersed-boundary finite-difference methods for three-dimensional
778 complex flow simulations, Journal of Computational Physics 161 (2000)
779 35–60.
- 780 [23] Y. H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method
781 for flow in complex geometry, Journal of Computational Physics 192
782 (2003) 593–623.
- 783 [24] E. Balaras, Modeling complex boundaries using an external force field

- 784 on fixed cartesian grids in large-eddy simulations, *Computers & Fluids*
785 33 (2004) 375–404.
- 786 [25] M. Kara, T. Stoesser, R. McSherry, Calculation of fluid-structure inter-
787 action: methods, refinements, applications, *Proceedings of the Institu-*
788 *tion of Civil Engineers. Engineering and Computational Mechanics* 168
789 (2015) 59–78.
- 790 [26] H. S. Udaykumar, W. Shyy, M. M. Rao, Elafint: A mixed eulerian-
791 lagrangian method for fluid flows with complex and moving boundaries,
792 *International Journal for Numerical Methods in Fluids* 22 (1996) 691–
793 712.
- 794 [27] T. Ye, R. Mittal, H. S. Udaykumar, W. Shyy, An accurate cartesian
795 grid method for viscous incompressible flows with complex immersed
796 boundaries, *Journal of Computational Physics* 156 (1999) 209–240.
- 797 [28] P. G. Tucker, Z. Pan, A cartesian cut cell method for incompressible
798 viscous flow, *Applied Mathematical Modelling* 24 (2000) 591–606.
- 799 [29] D. M. Ingram, D. M. Causon, C. G. Mingham, Developments in carte-
800 sian cut cell methods, *Mathematics and Computers in Simulation* 61
801 (2003) 561–572.
- 802 [30] M. P. Kirkpatrick, S. W. Armfield, J. H. Kent, A representation of
803 curved boundaries for the solution of the navier-stokes equations on a
804 staggered three-dimensional cartesian grid, *Journal of Computational*
805 *Physics* 184 (2003) 1–36.

- 806 [31] M. H. Chung, Cartesian cut cell approach for simulating incompressible
807 flows with rigid bodies of arbitrary shape, *Computers & Fluids* 35 (2006)
808 607–623.
- 809 [32] P. Lin, A fixed-grid model for simulation of a moving body in free surface
810 flows, *Computers & Fluids* 36 (2007) 549–561.
- 811 [33] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method
812 for wave-body interactions, *Journal of Computational Physics* 228
813 (2009) 6590–6616.
- 814 [34] H. Johansen, P. Colella, A cartesian grid embedded boundary method
815 for poisson’s equation on irregular domains, *Journal of Computational*
816 *Physics* 147 (1998) 60–85.
- 817 [35] D. Calhoun, R. J. LeVeque, A cartesian grid finite-volume method
818 for the advection-diffusion equation in irregular geometries, *Journal of*
819 *Computational Physics* 157 (2000) 143–180.
- 820 [36] M. Popescu, R. Vedder, W. Shyy, A finite volume-based high-order,
821 cartesian cut-cell method for wave propagation, *International Journal*
822 *for Numerical Methods in Fluids* 56 (2008) 1787–1818.
- 823 [37] L. Duan, X. W. Wang, X. L. Zhong, A high-order cut-cell method
824 for numerical simulation of hypersonic boundary-layer instability with
825 surface roughness, *Journal of Computational Physics* 229 (2010) 7207–
826 7237.
- 827 [38] Q. H. Liang, J. Zang, A. G. L. Borthwick, P. H. Taylor, Shallow flow

- 828 simulation on dynamically adaptive cut cell quadtree grids, *International*
829 *Journal for Numerical Methods in Fluids* 53 (2007) 1777–1799.
- 830 [39] K. Fidkowski, D. Darmofal, A triangular cut-cell adaptive method for
831 high-order discretizations of the compressible navierstokes equations,
832 *Journal of Computational Physics* 225 (2007) 1653–1672.
- 833 [40] H. Ji, F. S. Lien, E. Yee, Numerical simulation of detonation using an
834 adaptive cartesian cut-cell method combined with a cell-merging tech-
835 nique, *Computers & Fluids* 39 (2010) 1041–1057.
- 836 [41] D. Hartmann, M. Meinke, W. Schroder, A strictly conservative carte-
837 sian cut-cell method for compressible viscous flows on adaptive grids,
838 *Computer Methods in Applied Mechanics and Engineering* 200 (2011)
839 1038–1052.
- 840 [42] B. Muralidharan, S. Menon, A high-order adaptive cartesian cut-cell
841 method for simulation of compressible viscous flow over immersed bod-
842 ies, *Journal of Computational Physics* 321 (2016) 342–368.
- 843 [43] M. Meyer, A. Devesa, S. Hickel, X. Y. Hu, N. A. Adams, A conservative
844 immersed interface method for large-eddy simulation of incompressible
845 flows, *Journal of Computational Physics* 229 (2010) 6300–6317.
- 846 [44] Xie, Z., Lin, B., Falconer, R.A., Large-eddy simulation of the turbu-
847 lent structure in compound open-channel flows, *Advances in Water*
848 *Resources* 53 (2013) 66–75.
- 849 [45] Xie, Z., Lin, B., Falconer, R.A., Maddux, T.B., Large-eddy simulation

- 850 of turbulent open-channel flow over three-dimensional dunes, *Journal of*
851 *Hydraulic Research* 51 (2013) 494–505.
- 852 [46] F. Gao, D. M. Ingram, D. M. Causon, C. G. Mingham, The development
853 of a cartesian cut cell method for incompressible viscous flows, *International*
854 *Journal for Numerical Methods in Fluids* 54 (2007) 1033–1053.
- 855 [47] W. Bai, C. G. Mingham, D. M. Causon, L. Qian, Finite volume simula-
856 tion of viscous free surface waves using the cartesian cut cell approach,
857 *International Journal for Numerical Methods in Fluids* 63 (2010) 69–95.
- 858 [48] T. Q. Li, P. Troch, J. De Rouck, Wave overtopping over a sea dike,
859 *Journal of Computational Physics* 198 (2004) 686–726.
- 860 [49] Z. Xie, Two-phase flow modelling of spilling and plunging breaking
861 waves, *Applied Mathematical Modelling* 37 (2013) 3698–3713.
- 862 [50] Z. Xie, Numerical modelling of wind effects on breaking solitary waves,
863 *European Journal of Mechanics - B/Fluids* 43 (2014) 135–147.
- 864 [51] S. T. Dang, E. A. Meese, J. C. Morud, S. T. Johansen, Numerical
865 approach for generic three-phase flow based on cut-cell and ghost fluid
866 methods, *International Journal for Numerical Methods in Fluids* 91
867 (2019) 419–447.
- 868 [52] O. Desjardins, J. O. McCaslin, M. Owkes, , P. Brady, Direct numerical
869 and large-eddy simulation of primary atomization in complex geome-
870 tries, *Atomization and Sprays* 23 (2013) 1001–1048.

- 871 [53] Xie, Z., Lin, B., Falconer, R.A., Turbulence characteristics in
872 free-surface flow over two-dimensional dunes, *Journal of Hydro-*
873 *environmental Research* 8 (2014) 200–209.
- 874 [54] P. McCorquodale, P. Colella, H. Johansen, A cartesian grid embedded
875 boundary method for the heat equation on irregular domains, *Journal*
876 *of Computational Physics* 173 (2001) 620–635.
- 877 [55] D. M. Causon, D. M. Ingram, C. G. Mingham, A cartesian cut cell
878 method for shallow water flows with moving boundaries, *Advances in*
879 *Water Resources* 24 (2001) 899–911.
- 880 [56] G. Dechrste, L. Mieussens, A cartesian cut cell method for rarefied flow
881 simulations around moving obstacles, *Journal of Computational Physics*
882 314 (2016) 465–488.
- 883 [57] W. P. Bennett, N. Nikiforakis, R. Klein, A moving boundary flux sta-
884 bilization method for cartesian cut-cell grids using directional operator
885 splitting, *Journal of Computational Physics* 368 (2018) 333–358.
- 886 [58] M. Meinke, L. Schneiders, C. Gunther, W. Schroder, A cut-cell method
887 for sharp moving boundaries in cartesian grids, *Computers & Fluids* 85
888 (2013) 135–142.
- 889 [59] L. Schneiders, D. Hartmann, M. Meinke, W. Schroder, An accurate
890 moving boundary formulation in cut-cell methods, *Journal of Compu-*
891 *tational Physics* 235 (2013) 786–809.
- 892 [60] B. Muralidharan, S. Menon, Simulation of moving boundaries inter-
893 acting with compressible reacting flows using a second-order adaptive

- 894 cartesian cut-cell method, *Journal of Computational Physics* 357 (2018)
895 230–262.
- 896 [61] H. S. Udaykumar, H. C. Kan, W. Shyy, R. TranSonTay, Multiphase
897 dynamics in arbitrary geometries on fixed cartesian grids, *Journal of*
898 *Computational Physics* 137 (1997) 366–405.
- 899 [62] M. H. Chung, An adaptive cartesian cut-cell/level-set method to simu-
900 late incompressible two-phase flows with embedded moving solid bound-
901 aries, *Computers & Fluids* 71 (2013) 469–486.
- 902 [63] L. Qian, D. M. Causon, C. G. Mingham, D. M. Ingram, A free-surface
903 capturing method for two fluid flows with moving bodies, *Proceedings*
904 *of the Royal Society a-Mathematical Physical and Engineering Sciences*
905 462 (2006) 21–42.
- 906 [64] W. Benguigui, A. Doradoux, J. Lavieville, S. Mimouni, E. Longatte, A
907 discrete forcing method dedicated to moving bodies in twophase flow,
908 *International Journal for Numerical Methods in Fluids* 88 (2018) 315–
909 333.
- 910 [65] Z. Z. Hu, D. M. Causon, C. G. Mingham, L. Qian, A cartesian cut cell
911 free surface capturing method for 3d water impact problems, *Interna-*
912 *tional Journal for Numerical Methods in Fluids* 71 (2013) 1238–1259.
- 913 [66] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski,
914 B. Buchner, A volume-of-fluid based simulation method for wave impact
915 problems, *Journal of Computational Physics* 206 (2005) 363–393.

- 916 [67] Q. Chen, J. Zang, A. S. Dimakopoulos, D. M. Kelly, C. J. K. Williams, A
 917 cartesian cut cell based two-way strong fluid-solid coupling algorithm for
 918 2d floating bodies, *Journal of Fluids and Structures* 62 (2016) 252–271.
- 919 [68] L. Schneiders, C. Gunther, M. Meinke, W. Schroder, An efficient con-
 920 servative cut-cell method for rigid bodies interacting with viscous com-
 921 pressible flows, *Journal of Computational Physics* 311 (2016) 62–86.
- 922 [69] P. Heinrich, Nonlinear water waves generated by submarine and aerial
 923 landslides, *Journal of Waterway, Port, Coastal, and Ocean Engineering-*
 924 *ASCE* 118 (1992) 249–266.
- 925 [70] T. R. Wu, A numerical study of three-dimensional breaking waves and
 926 turbulence effects, Ph.D. thesis, Cornell University, 2004.
- 927 [71] J. Smagorinsky, General circulation experiments with the primitive
 928 equations: I. the basic experiment, *Monthly Weather Review* 91 (1963)
 929 99–164.
- 930 [72] D. K. Lilly, A proposed modification of the germano-subgrid-scale clo-
 931 sure method, *Physics of Fluids A* 4 (1992) 633–635.
- 932 [73] C. Hirsch, Numerical computation of internal and external flows intro-
 933 duction to the fundamentals of CFD, new ed., Butterworth-Heinemann,
 934 Oxford, 2007.
- 935 [74] Z. Xie, Numerical study of breaking waves by a two-phase flow model,
 936 *International Journal for Numerical Methods in Fluids* 70 (2012) 246–
 937 268.

- [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.

- 960 [84] N. P. Waterson, H. Deconinck, Design principles for bounded higher-
 961 order convection schemes - a unified approach, *Journal of Computational*
 962 *Physics* 224 (2007) 182–207.
- 963 [85] S. V. Patankar, *Numerical heat transfer and fluid flow*, Taylor & Fran-
 964 cis,, London, 1980.
- 965 [86] J. C. Martin, W. J. Moyce, An experimental study of the collapse of
 966 liquid columns on a rigid horizontal plane .4., *Philosophical Transac-*
 967 *tions of the Royal Society of London Series a-Mathematical and Physical*
 968 *Sciences* 244 (1952) 312–324.
- 969 [87] D. Pavlidis, J. L. M. A. Gomes, Z. Xie, J. R. Percival, C. C. Pain,
 970 O. K. Matar, Compressive advection and multi-component methods
 971 for interface-capturing, *International Journal of Numerical Methods in*
 972 *Fluids* 80 (2016) 256–282.
- 973 [88] D. Tritton, Experiments on the flow past a circular cylinder at low
 974 reynolds number, *Journal of Fluid Mechanics* 6 (1959) 547–567.
- 975 [89] B. Fornberg, A numerical study of steady viscous flow past a circular
 976 cylinder, *Journal of Fluid Mechanics* 98 (1980) 819–855.
- 977 [90] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface carte-
 978 sian grid method i: An easily implemented technique for 3d moving
 979 boundary computations, *Journal of Computational Physics* 210 (2005)
 980 1–31.

- 981 [91] A. A. Shirgaonkar, M. A. MacIver, N. A. Patankar, A new mathemat-
 982 ical formulation and fast algorithm for fully resolved simulation of self-
 983 propulsion, *Journal of Computational Physics* 228 (2009) 2366–2390.
- 984 [92] J. Wu, C. Shu, Y. H. Zhang, Simulation of incompressible viscous flows
 985 around moving objects by a variant of immersed boundary lattice boltz-
 986 mann method, *International Journal of Numerical Methods in Fluids*
 987 62 (2010) 327–354.
- 988 [93] I. M. Janosi, D. Jan, K. G. Szabo, T. Tel, Turbulent drag reduction in
 989 dam-break flows, *Experiments in Fluids* 37 (2004) 219–229.
- 990 [94] Greenhow, M., Moyo, S., Water entry and exit of horizontal circular
 991 cylinders, *Philosophical Transactions of the Royal Society of London.*
 992 *Series A: Mathematical, Physical and Engineering Sciences* 355 (1997)
 993 551–563.
- 994 [95] P. L. F. Liu, T. R. Wu, F. Raichlen, C. E. Synolakis, J. C. Borrero,
 995 Runup and rundown generated by three-dimensional sliding masses,
 996 *Journal of Fluid Mechanics* 536 (2005) 107–144.
- 997 [96] Z. Xie, A two-phase flow model for three-dimensional breaking waves
 998 over complex topography, *Proceedings of the Royal Society A: Mathe-*
 999 *matical, Physical & Engineering Sciences* 471 (2015) 20150101.