A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models

Original
A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models / Tafuni, A.; Domínguez, J. M.; Vacondio, R.; Crespo, A. J. C.. - In: COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING. - ISSN 0045-7825. - 342:(2018), pp. 604-624. [10.1016/j.cma.2018.08.004]

Availability:
This version is available at: 11381/2850222 since: 2021-10-13T16:18:44Z

Publisher:
Elsevier B.V.

Published
DOI:10.1016/j.cma.2018.08.004

A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models

A. Tafuni, J.M. Domínguez, R. Vacondio, A.J.C. Crespo

PII: S0045-7825(18)30390-6
DOI: https://doi.org/10.1016/j.cma.2018.08.004
Reference: CMA 12018

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date : 28 March 2018
Revised date : 2 August 2018
Accepted date : 6 August 2018

Please cite this article as: A. Tafuni, J.M. Domínguez, R. Vacondio, A.J.C. Crespo, A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models, *Comput. Methods Appl. Mech. Engrg.* (2018), https://doi.org/10.1016/j.cma.2018.08.004

A versatile algorithm for the treatment of open boundary conditions in Smoothed particle hydrodynamics GPU models

A. Tafuni[a,*], J. M. Domínguez[b], R. Vacondio[c], A. J. C. Crespo[b]

[a] *New York University - 6 MetroTech Center, Brooklyn NY, 11201 USA*
[b] *University of Vigo - Campus As Lagoas, Ourense, 32004 Spain*
[c] *University of Parma - Parco Area delle Scienze, 181/A, Parma, 43124 Italy*

Abstract

An open boundary algorithm for weakly compressible Smoothed particle hydrodynamics (WCSPH) numerical models is presented. Open boundary conditions are implemented by means of buffer regions whereby physical quantities are either imposed or extrapolated from the fluid region using a first-order accurate SPH interpolation. A unique formulation has been developed which can be used for inflow, outflow, and mixed open boundary conditions. The extrapolation process from the fluid domain encompasses quantities such as velocity, density, pressure and also free-surface elevation. The algorithm has been parallelized for both CPU and general-purpose on graphics processing units (GPGPU) and it has been tested against the 2-D reference solutions of flow past a cylinder and open channel flow. Finally, its capability to simulate 2-D and 3-D complex flows such as water waves and flow past a surface-piercing extraterrestrial submarine is demonstrated.

Keywords: SPH, inlet, outlet, open boundary, free surface, CFD
2010 MSC: 76M28, 65Y05, 65Z05

---

1. Introduction

Smoothed particle hydrodynamics (SPH) is a numerical method originally developed for astrophysical modeling [1, 2] and later adapted for free-surface

---

[*] Corresponding author
Email address: `atafuni@nyu.edu` (A. Tafuni)

flow simulations [3, 4]. In recent years, the use of SPH as a predictive tool has become more significant due to its application to several different industrial and environmental problems [5, 6]. Capabilities such as simulating interface flows, strong nonlinearities, and fluid-structure interactions in the presence of moving objects are some of the reasons behind its effectiveness as a computational fluid dynamics (CFD) technique [7, 8, 9, 10, 11]. However, despite the increasing success of SPH numerical schemes, there are different areas where further development is still required in order to increase the number of applications that can benefit from a SPH approach, see [12, 13] for a comprehensive review.

One of such areas is represented by the discretization of boundary conditions, which due to the Lagrangian nature of the SPH numerical scheme are more complicated than in established Eulerian approaches. Particularly, inflow and outflow boundary conditions represent a necessity in various fluid dynamics simulations in order to limit the size of the computational domain to a region of interest while avoiding spurious oscillations that can compromise the simulation accuracy. Robust open boundaries are thus a required tool to simulate several engineering and environmental problems, such as fluid flow in rivers and channels. They are also essential for coupling SPH schemes with other numerical models [14, 15].

The topic of open boundaries in SPH has been previously addressed by different authors. A non-reflecting open boundary formulation for internal flows has been proposed by Lastiwka et al. in [16], adopting Riemann invariants and, more recently, by Alvarado-Rodríguez et al. in [17], adopting a different formulation based on an anisotropic wave equation for the velocity field at the outlet. Vacondio et al. [18] introduced open boundary conditions in the framework of an SPH model for shallow water equations, also adopting Riemann invariants. Federico et al. [19] presented an implementation of open boundary conditions for weakly compressible SPH scheme suitable for free-surface flow. All different approaches cited above are based on the creation of buffer layers for inflow/outflow regions where buffer particles are created and deleted. Ferrand et al. [20] introduced a different approach based on the generalization of the semi-analytical boundary conditions method to impose unsteady open boundaries in a weakly compressible SPH model. Furthermore, different open boundary formulations in the framework of incompressible SPH (ISPH) models have been proposed in literature [21, 22, 23].

The main objective of the present work is to develop robust and accurate open boundary conditions that can be implemented in a weakly compressible

SPH parallel solver to simulate real engineering problems with free-surface flows. The algorithm has been developed in the framework of the established open-source solver DualSPHysics [24], which incorporates modeling capabilities for both CPU and GPU computing. The code encompasses some of the most advanced and well known optimizations of the CUDA architecture, as well as GPU optimizations that are relevant to the SPH method, see [25] for more details. The code has also been validated for use on more than one device (multi-GPU SPH) [26], allowing simulations of free-surface problems with a total number of particles exceeding 10^9 on just 64 graphics cards.

The approach based on buffer layers has been herein adopted because it can be efficiently parallelized on hybrid architectures and applied to both 2-D and 3-D simulations. Buffer regions prevent errors generated by the kernel truncation near the boundaries and particles inside the buffer areas are created and/or deleted to prevent the formation of voids. To enforce flow conditions at the boundaries, velocity and/or pressure can be assigned to the particles inside the buffer region. The algorithm is developed in such a way that physical information of the buffer particles can also be extrapolated from the fluid domain using a first order consistent procedure based on ghost points located in the fluid domain near the boundary. Additionally, a methodology to extrapolate (or impose) time-varying free-surface elevation in the buffer regions has been developed, allowing the generation of free-surface waves while minimizing the reflection of numerical noise into the fluid.

This unique open boundary condition formulation can be used to impose inflow/outflow boundary conditions according to the different physical quantities imposed and/or extrapolated from the flow field. The process of extracting physical quantities from the inside of the fluid domain to the buffer particles allows to convey time-dependent pressure fluctuations and 2-D/3-D vortex structures across the boundaries, preventing disturbances from diffusing into the fluid. One of the novelties of the proposed algorithm is that there is no distinction between an inflow and an outflow buffer since the same class of particles is used in both cases. A buffer layer can be both inlet and outlet depending on the type of problem to be solved, and thus on how the properties of these particles are calculated. Another novel aspect is represented by the ability of the present formulation to enforce unsteady velocity and pressure profiles and/or pressure and velocity gradients along any given direction in the buffer regions, as well as unsteady free-surface elevation.

The manuscript is organized as follows: after a brief review of the main principles behind SPH interpolations in Section 2.1 and a summary of the weakly compressible SPH (WCSPH) equations in Sections 2.2 and 2.3, the proposed open boundary algorithm is discussed in detail in Section 3. Subsequently, open boundary conditions are employed to simulate flow past a circular cylinder at different Reynolds numbers in Section 4.1, open channel flow with a convergence analysis in Section 4.2 and wave generation in Section 4.3. Finally, Section 4.4 presents an application of the algorithm to model the 3-D flow past the Titan submarine, designed by NASA to navigate the liquid hydrocarbon seas of Saturn's largest moon. Some conclusions and future work are drawn in Section 5.

2. SPH formalism

Smoothed particle hydrodynamics is one of the most advanced particle methods in numerical hydrodynamics, particularly suited for the simulation of flow with a free surface and characterized by large gradients. In the next subsections a brief description of the SPH formalism and properties is presented, as well as the most relevant equations for WCSPH.

2.1. Mathematical Background

The discretization of the computational domain in SPH is made through a set of Lagrangian points, identified as particles. Each particle has material properties (e.g. velocity, density, etc.) and retains flow information at its location in the computational space. The particles also act as interpolation points at which the convolution of a field function evaluated at neighboring particles with a smooth interpolant (kernel) provides an approximation of that same function at the target particle. The two SPH interpolation steps performed on a generic function, $f(\mathbf{x})$, are provided below:

$$\langle f(\mathbf{x})\rangle \triangleq \int_{\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)\,d\mathbf{x}' \tag{1}$$

$$\langle f(\mathbf{x}_k)\rangle = \sum_{l=1}^{N} \frac{m_l}{\rho_l} f(\mathbf{x}_l)W_{k,l} \tag{2}$$

where $\mathbf{x}$ is the target position vector, $\mathbf{x}'$ represents the position vector of a generic particle located within the kernel support Ω, N is the total number of particles inside the kernel domain centered at particle k, m_l and ρ_l are the mass and density of the interpolating particle l, and W is the kernel function, with $(\cdot)_{k,l} = (\cdot)_k - (\cdot)_l$. The angular brackets denote the SPH approximation. The parameter h is the smoothing length, controlling influence of neighboring particles in the computation of $f(\mathbf{x})$. A simple substitution of the generic function $f(\mathbf{x})$ with the density function $\rho(\mathbf{x})$, leads to the SPH density estimate for particle k, $\rho(\mathbf{x}_k)$, given as:

$$\rho(\mathbf{x}_k) = \sum_{l=1}^{N} m_l W_{k,l} \tag{3}$$

which constitutes the starting point for deriving a set of conservation equations in SPH, presented in the next section.

2.2. Conservation Equations and Pressure Treatment

In the present work SPH is used to simulate free-surface flow with small characteristic Mach number, thus compressibility effects are negligible. The continuity and Navier-Stokes equations in Lagrangian form for a weakly-compressible fluid are:

$$\frac{d\rho}{dt} + \rho \nabla \cdot \mathbf{u} = 0 \tag{4}$$

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla P + \mathbf{g} + \frac{1}{\rho}\nabla \cdot \tau \tag{5}$$

where $d(\cdot)/dt = \partial(\cdot)/\partial t + \mathbf{v} \cdot \nabla(\cdot)$, $\nabla\ (\cdot)$ is the gradient operator, $\nabla \cdot (\cdot)$ is the divergence operator, $\mathbf{u}$ is the velocity vector, P is the pressure, ρ is the fluid density, $\mathbf{g}$ is gravity, and τ is the deviatoric component of the total stress tensor. Using the kernel and particle approximations in Equations (1) and (2), a set of conservation equations for WCSPH is obtained as follows:

$$\frac{d\rho_k}{dt} = \sum_{l=1}^{N} m_l \mathbf{u}_{kl} \cdot \nabla_k W_{k,l} \tag{6}$$

$$\frac{\mathrm{d}\mathbf{u}_k}{\mathrm{d}t} = -\sum_{l=1}^{N} m_l \left[\left(\frac{P_l + P_k}{\rho_l\ \rho_k} \right) \nabla_k W_{k,l} - \Pi_{k,l} \right] + \mathbf{g} \tag{7}$$

To provide a closure relation for pressure the following equation of state is employed:

$$P_k = \varphi(\kappa_k^\gamma - 1) \tag{8}$$

with $\kappa_k = \rho_k/\rho_0$, γ being a coefficient and ρ_0 being the fluid reference density. It is commonly accepted to adopt the values $\gamma = 7$ and $\rho_0 = 10^3$ kg/m^3 as optimal values for water simulations with WCSPH algorithms [27]. The coefficient φ controls the density variations within the limits imposed by stability criteria and is function of the numerical speed of sound $c_0 = c(\rho_0)$, as explained in more details in [28, 29]. The term $\Pi_{k,l}$ in Equation (7) accounts for viscous and turbulent stresses and an expression is given by:

$$\Pi_{k,l} = \left[\left(\frac{4\nu_0 \mathbf{x}_{k,l} \cdot \nabla_k W_{k,l}}{(\rho_k + \rho_l)(\mathbf{x}_{k,l}^2 + \eta^2)} \right) \mathbf{u}_{k,l} + \left(\frac{\tau_l^*}{{\rho_l}^2} + \frac{\tau_k^*}{{\rho_k}^2} \right) \nabla_k W_{k,l} \right] \tag{9}$$

where $\eta = 0.1\ h$ is used to avoid singularities in the denominator. Both laminar and turbulent stresses are considered. The laminar portion is given by the first term in Equation (9), with $\nu_0 = \mu/\rho_0$ being the fluid kinematic viscosity, and is based on the discretization proposed by Lo and Shao [30]. Conversely, a large-eddy simulation (LES) approach is used to model the turbulent stresses, τ, in the second term of Equation (9), as proposed in [31].

2.3. Other Relevant SPH Aspects

DualSPHysics includes the capability of activating a corrective term in the continuity equation to weaken the high-frequency low-amplitude oscillations affecting WCSPH density fields due to the natural disorder of the Lagrangian particles. This diffusive term is added to the right-hand side of Equation (6) and has the form:

$$2\delta h c_0 \sum_{l=1}^{N} m_l \left(1 - \frac{\rho_k}{\rho_l}\right) \frac{\mathbf{x}_{k,l} \cdot \nabla_k W_{k,l}}{|\mathbf{x}_{k,l}|^2} \tag{10}$$

This represents the original δ-SPH term formulated in [32], with the parameter δ used to tune the intensity of the diffusion (inactive when $\delta = 0$).

A second important aspect is anisotropic particle spacing, a critical stability issue in SPH as particles cannot maintain a uniform distribution in certain flow conditions. The result is the introduction of noise in both the velocity and pressure fields, as well as the creation of voids in certain areas of the fluid domain. To address this issue, Xu et al. [33] have proposed a particle shifting algorithm. The algorithm has firstly been created for incompressible SPH, but can be extended to WCSPH models, as done by Vacondio et al. in [34, 35]. The shifting correction forces the movement of particles towards areas with lower particle concentrations, allowing the domain to maintain a near-uniform particle distribution and eliminating any voids that may occur. An improvement on the initial shifting algorithm has been proposed by Lind et al. [36], where Fick's first law of diffusion is used to control the shifting magnitude and direction. Assuming that the flux, i.e. the number of particles passing through a unit surface per unit time, is proportional to the velocity of the particles, a shifting velocity and subsequently a particle shifting distance, $\delta\mathbf{r}$, can be modeled as:

$$\delta\mathbf{r} = -D\nabla C \tag{11}$$

where D is the diffusion coefficient that controls the shifting magnitude and absorbs the constants of proportionality, and ∇C is the gradient of the particle concentration. The latter is found using the SPH gradient operator, whereas D is computed following the approach proposed in Skillen et al. [37], wherein $D = Ah|\mathbf{u}|_2$. A is a dimensionless constant that is independent of the problem setup and discretization, h is the smoothing length, and $|\cdot|_2$ indicates the 2-norm operator. The effectiveness of the shifting algorithm is strongly dependent on the kernel having a full support, and this clearly poses an issue in the vicinity of the free surface, where the kernel is truncated. The correction proposed in [36] limits the diffusion in the direction normal to the free surface, while allowing shifting in the direction tangent to the free surface. To check if a particle is in the vicinity of the free surface, the divergence of its position vector is calculated as:

$$\nabla\cdot\mathbf{r} = \sum_{l=1}^{N} \frac{m_l}{\rho_l}\mathbf{r}_{k,l}\cdot\nabla_k W_{k,l} \tag{12}$$

and the result is compared with the threshold value of $\nabla\cdot\mathbf{r}$ at the free surface, A_T. In the present work A_T is assumed equal to 1.5 in 2-D and 2.5 in 3-D, as suggested in Lee at al. [38], Lind et al. [36], and Mokos et al.

[39]. If $\nabla \cdot \mathbf{r} \leq A_T$, the shifting distance in Equation (11) is multiplied by a correction coefficient, A_{FS}, written as:

$$A_{FS} = \frac{\nabla \cdot \mathbf{r} - A_T}{A_{FK} - A_T} \tag{13}$$

where A_{FK} is the value of the divergence of the position in the case of a full kernel support, given by $A_{FK} = 2$ in 2-D and $A_{FK} = 3$ in 3-D.

3. Open Boundary Algorithm Rationale

Several types of boundary conditions are already available in DualSPHysics to simulate a variety of engineering applications. Some examples are the dynamic boundary particles [40], floating bodies [7], SPH-DEM coupling [8], and periodic boundary conditions [41]. Nevertheless, none of these formulations is appropriate when the computational problem requires specific open boundary conditions to be enforced at the domain edges. The classic example of flow past an object is one of such cases, where usually an inflow velocity needs to be prescribed at the inlet while other velocity or pressure conditions can be either prescribed or extracted from the fluid domain at the outlet. To address this issue, an open boundary algorithm has been implemented in DualSPHysics. The sketch in Figure 1 briefly summarizes the working principles of the algorithm in the generic case of a fluid flowing near a buffer area identifying an open boundary. The innermost dashed curve represents the buffer threshold boundary, i.e. the fluid-buffer interface, followed by a layer of SPH particles used to enforce certain boundary conditions. The buffer width is chosen to equal or exceed the kernel radius so to ensure full kernel support for the fluid particles in the near proximity of an inlet or outlet.

Two ways of providing the information to an open boundary are considered: physical quantities are either assigned a priori or extrapolated from the fluid domain to the buffer zones (inflow and outflow) using ghost nodes. A similar idea is used in [42] to enforce closed boundary conditions. As can be seen in Figure 1, the position of the ghost nodes is obtained by mirroring the boundary particles into the fluid along a direction that is normal to the open boundary. In order to calculate fluid quantities at the ghost nodes, a standard particle interpolation would not be consistent due to the proximity of these points to an open boundary, which translates into the lack of a full kernel support. The method proposed by Liu and Liu [43] is thereby adopted to retrieve first order kernel and particle consistency. The multi-dimensional first-order Taylor series approximation of the field function $f(\mathbf{x})$ multiplied by the kernel function evaluated at particle k, $W_k(\mathbf{x})$, and its first order derivatives, $W_{k,\beta}(\mathbf{x})$, are given by:

$$\int f(\mathbf{x})W_k(\mathbf{x})\mathrm{d}\mathbf{x} = f_k \int W_k(\mathbf{x})\mathrm{d}\mathbf{x} + f_{k,\beta}\int(\mathbf{x}-\mathbf{x}_k)W_k(\mathbf{x})\mathrm{d}\mathbf{x} \tag{14}$$

$$\int f(\mathbf{x})W_{k,\beta}(\mathbf{x})\mathrm{d}\mathbf{x} = f_k \int W_{k,\beta}(\mathbf{x})\mathrm{d}\mathbf{x} + f_{k,\beta}\int(\mathbf{x}-\mathbf{x}_k)W_{k,\beta}(\mathbf{x})\mathrm{d}\mathbf{x} \tag{15}$$

Here β is an index going from 1 to d, the total number of dimensions. Equations (14) and (15) form a system of $d+1$ equations in $d+1$ unknowns, i.e. f_k and $f_{k,\beta}$. Using particle notation, the solution to this system is found as:

$$f_k = \frac{\begin{vmatrix} \sum_l f_l W_{kl}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl}\Delta V_l \\ \sum_l f_l W_{kl,\beta}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl,\beta}\Delta V_l \end{vmatrix}}{\begin{vmatrix} \sum_l f(\mathbf{x}) W_{kl}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl}\Delta V_l \\ \sum_l f(\mathbf{x}) W_{kl,\beta}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl,\beta}\Delta V_l \end{vmatrix}} \tag{16}$$

$$f_{k,\beta} = \frac{\begin{vmatrix} \sum_l W_{kl}\Delta V_l & \sum_l f_l W_{kl}\Delta V_l \\ \sum_l W_{kl,\beta}\Delta V_l & \sum_l f_l W_{kl,\beta}\Delta V_l \end{vmatrix}}{\begin{vmatrix} \sum_l W_{kl}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl}\Delta V_l \\ \sum_l W_{kl,\beta}\Delta V_l & \sum_l (\mathbf{x}_l-\mathbf{x}_k)W_{kl,\beta}\Delta V_l \end{vmatrix}} \tag{17}$$

These have been employed to find the value of f_o at the open boundary given the corrected values of f_k and $f_{k,\beta}$ at the ghost nodes:

$$f_o = f_k + (\mathbf{r}_o - \mathbf{r}_k)\cdot \widetilde{\nabla} f_k \tag{18}$$

where $\widetilde{\nabla} f_k$ is the corrected gradient calculated at the ghost nodes.

As previously mentioned, one of the novelties of the proposed algorithm is that there is no distinction between an inflow and an outflow buffer since the same class of particles is used in both cases. A buffer layer can be both inlet and outlet depending on the type of problem to be solved, and thus on how the properties of these particles are calculated. For example, if the buffer has to be used as an inlet, a velocity and water depth can be specified while the pressure or density can be extrapolated from the fluid particles.

Figure 2 further illustrates the working principles of the open boundary algorithm. To define the reference surface for the placement of buffer particles and ghost nodes, a series of fixed points is created along a user-defined curve in 2-D or surface in 3-D. Buffer particles are created along the normal direction to the open boundary, starting from the fixed points and only for depths below the defined water level. The latter can be either user-defined or extrapolated from the ghost nodes using the divergence of the position shown in Equation (12). A ghost node is considered above the free surface if $\nabla \cdot \mathbf{r} \leq 1.5$ in 2-D and $\nabla \cdot \mathbf{r} \leq 2.5$ in 3-D. This procedure implicitly assumes that the water level computed inside the fluid domain via the ghost nodes is mirrored in the buffer zone. Buffer particles adjacent to the fluid domain are generated starting at half the value of the particle spacing from the fixed points, and continuing in the normal direction to the boundary until the full radius of the kernel function is covered. As shown in Figures 2b to 2d, when a buffer particle crosses the fluid-buffer interface it becomes a fluid particle and simultaneously a new boundary particle is initialized in the buffer at a position $\mathbf{r}_{new}$:

$$\mathbf{r}_{new} = \left[\mathbf{r}_{fluid} - (\mathbf{r}_{fluid} - \mathbf{r}_{fixed}) \cdot \mathbf{n}_{fixed} - L_b\right] \mathbf{r}_{fixed} \tag{19}$$

where $\mathbf{r}_{fluid}$ and $\mathbf{r}_{fixed}$ are respectively the positions of the buffer particle converted into a fluid one and that of the associated fixed point, L_b is the buffer length (usually equal to $2h$), and $\mathbf{n}_{fixed}$ is the unit vector at the fixed point normal to the fluid-buffer interface, always pointing inside the fluid domain. The process is similar for the transition of fluid particles into a buffer region: when a fluid particle crosses a fluid-buffer interface it becomes a buffer particle and follows the flow conditions specified in the buffer. In this case there is no creation of new particles following this transition. Finally, when a buffer particle crosses the domain edge it is discarded from the computational space. Figures 2c and 2d show how changes in liquid depth are tackled by the algorithm: the maximum height of the buffer area is bounded by the largest vertical coordinate of the fixed points. On the other hand, the water level can be either imposed by the user or extrapolated from the near fluid. The extrapolated water level is computed as the maximum depth of the fluid particles as interpolated at a distance $2h$ from the inlet limit.

When the water level increases, new buffer particles are created to reach the new water level (Figure 2c). Conversely, when the water level decreases, any buffer particle above the water level is removed (Figure 2d).

In addition to the most common use of the buffer regions for open boundaries in SPH, the new algorithm presents a number of novel features that make it attractive for use in real engineering problems. One of these is the capability of enforcing unsteady velocity and pressure profiles and/or pressure and velocity gradients along a given direction in the buffer areas. Another, as previously mentioned, is the ability to simulate variable free-surface elevation, an essential aspect in flow with a free surface because of the presence of waves entering and exiting the computational domain. Finally, an important feature that has been implemented is the dual behavior of buffer areas, whereby flow reversion is possible. When the velocity is extrapolated from the fluid domain it is also possible to handle mixed velocity fields, where some fluid particles are moving towards and some others are moving away the buffer region. This flexibility of the open boundary algorithm is particularly important when simulating flow with strong rotations or oscillating nature, such as pulsatile flow. The availability of the current algorithm on both the parallel CPU and GPU versions of DualSPHysics allows considerable speed-ups when the code runs on high-end graphics cards or CPU clusters. This is of utmost importance when simulating real-life engineering problems since it allows using a large number of particles to discretize and study high-resolution flow in complicated geometries within reasonable computational time.

4. Results and Discussion

4.1. Flow Past a Circular Cylinder

Fluid flow past a circular cylinder is investigated to test the effectiveness of the proposed algorithm. This problem presents many numerical and experimental solutions, see for example Liu et al. (1998) [44], Calhoun (2002) [45], Marrone et al. (2013) [42], Vacondio et al. (2013) [34], and is therefore suitable for benchmarking SPH results. A cylinder of diameter $D = 0.1$ m is centered at the origin of a 2-D Cartesian coordinate system $O(x, y)$, as shown in Figure 3. The circular object is discretized by dynamic boundary particles [24]. The cylinder is surrounded by a viscous fluid filling a computational domain of dimensions $20D \times 25D$. These dimensions are chosen in accordance with those in [34] to minimize boundary effects and avoid a cumbersome particle convergence study, which is instead left for the next study case. The fluid is initialized with a constant density, $\rho_\infty = 10^3$ kg/m^3, a constant x-velocity, $U_\infty = 1$ m/s and a null y-velocity. The Reynolds number is defined as $\mathrm{Re} = U_\infty D \nu^{-1}$, where ν is the kinematic viscosity in m^2/s, systematically varied to span a range of Reynolds numbers between 20 and 200. Two buffer areas are used to enforce inflow and outflow boundary conditions. The left buffer (Buffer 1) represents the inlet zone, having the same height as the fluid domain and a width of four particle layers. This is done to enforce full kernel support during the particle approximation. A Dirichlet boundary condition for the velocity is imposed in the inlet zone, with the y-velocity of the particles set to zero and the x-velocity set to the constant value U_∞. The right buffer (Buffer 2) has the same dimensions as Buffer 1, however the velocity of the outlet particles is obtained by first extrapolating the fluid velocity at the respective ghost nodes and then applying the linear correction described in Section 3. The same extrapolation process is utilized to compute the density, and hence pressure, of all buffer particles, thus no a priori density assignment is made anywhere at the open boundaries. The particle spacing is set to $\Delta x = 0.01D$, giving 100 boundary particles across the cylinder diameter and approximately 5×10^6 total SPH particles. The smoothing length is set equal to $h = 1.5\Delta x$. Particle shifting and delta-SPH are activated, the former being particularly important to obtain a near-uniform particle distribution in the wake, the latter used to weaken oscillations in the density field typical of the SPH method. As previously mentioned, the viscosity is varied to simulate different Reynolds numbers, particularly a steady case (Re = 20), a transitional case (Re = 50) and two unsteady cases (Re = 100 and Re = 200).

Figures 4 to 6 depict close-up contours of several flow field variables at the same time instants for all simulated cases. Particularly, Figure 4 shows the dimensionless velocity magnitude, $U^* = u(x/D, y/D)U_\infty^{-1}$, Figure 5 shows the dimensionless pressure, $P^* = 2P(x/D, y/D)\rho_\infty^{-1}U_\infty^{-2}$, and Figure 6 shows the dimensionless vorticity, $\zeta^* = \zeta(x/D, y/D)DU_\infty^{-1}$. Though the simulations are performed using a weakly compressible SPH formulation, it is noted how the pressure field obtained for different Reynolds numbers (see Figure 5) does not show any spurious oscillations. Moreover, the first order consistent extrapolation procedure based on the ghost nodes inside the fluid domain is able to convey von Karman vortices across the boundary without introducing any disturbance inside the fluid region, as noticed in Figure 6.

A steady smooth wake, symmetric about the x-axis, can be observed behind the cylinder for the case Re = 20. Stagnation areas can be identified upstream and downstream of the cylinder, where the fluid halts and recirculates, respectively. For Re = 50, the flow is transitioning into the unsteady regime, but is still in the steady and attached configuration. Contours in Figures 4b, 5b and 6b show a longer wake as opposed to the previous case, with the symmetry starting to break along the x-axis. The line of zero vorticity has now started to warp, indicating the imminent onset of flow instability. Finally, for the unsteady cases Re = 100 and Re = 200, an oscillatory wake behind the cylinder is observed and expected, with the formation of the classical von Karman street. Contours of velocity, pressure, and vorticity highlight the presence of periodic counter-rotating vortices shed behind the cylinder. As predicted by the theory, the frequency of shed vortices and their rotational intensity are dependent on the Reynolds number since, as Re increases, a larger number of vortex cores is spotted in the same domain area. All results are in very good agreement with those in Liu et al. (1998) [44], Calhoun (2002) [45], Marrone et al. (2013) [42], Vacondio et al. (2013) [34].

To better assess the quality of the results, the vortex shedding frequency, f, is calculated by measuring the period between the passage of a vortex core and a successive one at a given location and taking the inverse. Table 1 reports calculated Strouhal numbers, $\mathrm{St} = fDU_\infty^{-1}$, for SPH simulations herein and in other literature works. A very close agreement with the cited literature can be noted, indicating that the physics of this problem is well-captured by SPH with the present open boundary implementation.

Streamlines are also presented in Figure 7a for the case Re = 20 and Figure 7b for the case Re = 200. For the former, it is possible to observe a well-ordered flow with two symmetric counter-rotating vortices in the wake. Changes in the velocity sign allow to measure a separation angle of approximately 42° from the upstream stagnation point, an excellent match with the result of 43° in [42] and 45.5° in [45]. Moreover, the length of the recirculation bubble is estimated at $0.94D$, in very close agreement with the value of $0.91D$ in [45]. Conversely, the unsteady case presents only one vortex core in the process of being shed as the flow becomes oscillatory and obviously no steady values for separation angle and bubble length can be provided.

Figure 8 shows the drag and lift coefficients, $C_D = 2F_D\rho_\infty^{-1}U_\infty^{-2}D^{-1}$ and $C_L = 2F_L\rho_\infty^{-1}U_\infty^{-2}D^{-1}$, respectively, against the dimensionless time, $tU_\infty D^{-1}$. Here F_D and F_L are the drag and lift forces, respectively. The drag coefficient converges to a steady value of $C_D = 2.29$ for Re = 20 and $C_D = 1.46$ for Re = 200, whereas a null lift coefficient is observed and expected for the steady case, and $C_L = 0.693$ for the unsteady case. Once again, very close agreement is seen with the results in Liu et al. (1998) [44], Calhoun (2002) [45], Marrone et al. (2013) [42], Vacondio et al. (2013) [34]. For the unsteady cases, the values of the vortex shedding frequency calculated earlier are also corroborated by the frequency of the lift force signal.

Figure 9 depicts the dependence of the drag coefficient on the Reynolds number for all cases considered in this work and the results graphed in [46]. An excellent overall match is observed, with a slight discrepancy for the higher Reynolds numbers, most likely due to the lack of higher resolution. It is remarked that, for the case of flow past a cylinder, the value of the smoothing-length-based Reynolds number, $\mathrm{Re}_h = U_\infty h \nu^{-1}$, should be around the unity or less for proper outcome of SPH computation. Having kept the particle spacing fixed throughout the simulation campaign for simplicity, the $\mathrm{Re} = 200$ case corresponds to a value of $\mathrm{Re}_h \approx 3$, and therefore a slight overestimation of force coefficients may have resulted. Nevertheless the simulations outcomes are largely satisfactory and prove the effectiveness of the implemented algorithm without the need of further investigating particle resolution. Finally, values of the lift coefficient are also presented in tabular form in Table 2 for both cases $\mathrm{Re} = 100$ and $\mathrm{Re} = 200$.

4.2. Open Channel Flow

Free-surface flow in a two-dimensional channel is studied next. The availability of analytical solutions for this case allows to quantify any numerical errors and assess the quality of the implemented boundary condition algorithm. The channel, shown in Figure 10, has a depth $H = 1$ m, a length $L = 8H$, and is bounded by two buffer areas, similarly to the previous study case. Inflow and outflow boundary conditions are imposed on the left and on the right of the domain, respectively. The flow is gravity-dominated and presents a free surface, which requires a third flow variable to be determined at the open boundaries, the water depth. For both buffers the water depth is extrapolated from the fluid domain using the ghost nodes. At the bottom, a no-slip condition is enforced. Finally, the density of outflow particles is now assigned so that a hydrostatic pressure distribution is obtained and enforced at the outlet throughout the simulation.

The fluid is initialized with a constant density $\rho = 10^3$ kg/m^3, a null velocity in the y-direction, and a velocity in the x-direction given by the analytical solution [47]:

$$u(x,y) = u(y) = g\frac{\sin\alpha}{\nu}\left(yH - \frac{1}{2}y^2\right) \tag{20}$$

where $g = 9.81$ m/s^2, $\alpha = 5°$ is the chosen channel slope, and a value $\nu = 0.0534$ m^2/s is assigned to the kinematic fluid viscosity to obtain a depth-based Reynolds number equal to $\mathrm{Re} = U_{\mathrm{avg}}H\nu^{-1} = 100$. Equation (20) is also used to initialize the left and right buffers and to define the inlet velocity during the simulation, whereas particles in bottom buffer (Buffer 3) are given a null velocity in all directions. As can be seen from Figure 10, a perfectly horizontal channel is adopted rather than an inclined one for a simpler implementation. Therefore, in order to properly simulate the gravity driven flow due to the presence of a 5-degree slope, the gravity vector is defined as $\mathbf{g} = \{g_x, g_y\} = \{g\sin\alpha, -g\cos\alpha\}$.

A convergence study is carried out by choosing an initial value of the particle spacing equal to $2H/25$ and successively halving it until an acceptable match with the analytical solution is observed. Figure 11 presents the percent error between the analytical and numerical velocities as a function of the dimensionless depth $H^* = y/H$, calculated as:

$$\%\ \mathrm{Error} = \frac{|u_{\mathrm{SPH}}(x_0,y) - u(y)|}{U_{\mathrm{avg}}} \times 100 \tag{21}$$

where:

$$U_{\mathrm{avg}} = \frac{1}{H}\int_0^H u(y)dy = g\frac{\sin\alpha}{\nu}\frac{H^2}{3} \tag{22}$$

Different markers correspond to different particle spacings ($2H/25$, $H/25$, $H/50$, $H/100$), returning an average number of particles over the depth equal to 12.5, 25, 50, and 100, respectively. The smoothing length-to-particle spacing ratio is constant and equal to 1.5, and $\delta = 0.1$ is employed to activate δ-SPH. In all four cases the value of x_0 is taken in the proximity of the outlet to allow for the largest error propagation as the fluid travels the entire domain before being sampled. A steady convergence rate is observed with an error that systematically decreases as the particle resolution is increased. The case $H/100$ shows an overall error within 1% of the average analytical solution and is henceforth considered to obtain velocity and pressure profiles.

The normalized horizontal velocity $U^* = u(x,y)/U_{\max}$ is shown in Figure 12 at several horizontal positions for both SPH simulation with resolution $H/100$ and theoretical predictions. Here $U_{\max} = u(x,H) = u(H)$ and is calculated from Equation (20), with each snapshot retrieved at the end of the simulation. Four different x-positions are considered, $x/L = 0$, $x/L = 1/3$, $x/L = 2/3$, and $x/L = 1$. As the horizontal distance from the inlet region increases, an excellent agreement can still be observed between SPH and the theory. Specifically at the outflow, the farthest region from the inflow area, the velocity of buffer particles matches the analytical solution to a great extent. This proves the effectiveness of the implemented algorithm in simulating flow with open boundaries without introducing numerical noise in the velocity field.

Figure 13 shows the comparison of velocity contours for the analytical and numerical solutions at time 20 s. As expected, the velocity field is identical in each panel, corroborating the quantitative results above. Close-up contours of dimensionless pressure $P^* = 2P\rho^{-1}U_{\max}^{-2}$ are also presented in Figure 14 with a focus on the domain edges: the pressure distribution in the vicinity of the two buffer areas is shown at time $t = 20$ s. In both regions the algorithm is actively used to extrapolate specific flow quantities using the linear correction. No trace of noise or other instability can be seen, further highlighting the correct functioning of the implemented algorithm.

4.3. Wave Generation

The capability of the proposed open boundary algorithm of generating waves is hereby investigated. A numerical tank is simulated where regular waves are generated first by using a piston-type wavemaker, a common approach for wave generation in SPH, and then by using the new open boundary formulation. The computational domain is sketched in Figure 15, with an initial water depth of $d = 0.27$ m. Regular water waves of height $H = 0.1$ m, with period $T = 1.3$ s and wavelength $\lambda = 1.89$ m, are propagated along a 6-m-long tank with a 1:5-sloped dissipative beach at the end, in order to absorb the reflected waves and analyze only the incident waves. These regular waves belong to Stokes' second-order wave theory, hence numerical results can be compared with theoretical predictions. Several numerical gauges are placed throughout the computational domain. Three wave gauges are placed at the free surface, $a = 1$ meter apart from each other and $2a$ meters (WG 1), $3a$ meters (WG 2), and $4a$ meters (WG 3) away from the generation area.

These measure the instantaneous free-surface elevation, η, over time. Additionally, a fourth gauge, d_{v}, is placed $0.15a$ meters below WG 1 to measure the orbital velocities at that position. Altomare et al. (2017) [48] suggest that a good balance between simulation accuracy and computational time is achieved with a particle spacing equal to $H/10$. Therefore the particles are initially spaced 1 cm away, leading to a total number of 18,000 SPH particles. As for previous cases, the ratio of the smoothing length to the particle distance is set to 1.5, and a value of 0.1 is assigned to the δ-SPH coefficient.

Two different SPH simulations are performed; one case with a moving piston, created as a column of dynamic boundary particles with their motion imposed such that the desired waves are generated; the second case uses a buffer area to create the same waves by simultaneously imposing the velocity and the water depth in Buffer 1, while extrapolating the density from the fluid domain interior.

Figures 16 and 17 present theoretical and numerical results for assessing the correct generation and propagation of waves via the open boundary formulation proposed herein. Free surface elevations are shown in Figure 16; the results from both SPH simulations (piston and I/O) match the theoretical solution to a good extent, indicating that the waves are properly generated in both cases. Orbital velocities are then compared in Figure 17; although the horizontal velocity is slightly underestimated by SPH, results in both panels seem to provide a satisfactory accuracy. Therefore, the two numerical solutions are in good agreement, indicating that the buffer algorithm implemented in this study can generate and propagate waves with at least the same level of accuracy of the piston-type wavemaker.

A snapshot of the horizontal velocity contours (U_x) taken at $t = 7.40$ s is shown in Figure 18. The top panel depicts the two-dimensional flow field obtained with the use of the moving piston, whereas the bottom panel shows the same contour plot for the inlet/outlet formulation. It can be noticed how the same velocity patterns and free-surface profiles are observed when using the two approaches. Results in Figures 16 to 18 confirm the effectiveness of the proposed open boundary conditions algorithm for monochromatic waves. One of the advantages of using open boundaries to propagate waves into the SPH domain is the ease of coupling with other models, such as mesh-based codes or wave propagation models that can provide the velocity field or the depth time series to be imposed in the buffer zone to achieve correct inlet/outlet behavior for wave generation. A coupling approach based on the presented algorithm can be an alternative to, for example, the one used in

[49], where the boundary particles of a piston move according to the velocity provided by another model, resulting in deformation and displacement of the piston in long simulations.

4.4. Flow Past the Titan Submarine

The last case presented is the study of the three-dimensional flow around the hull of the NASA Phase 1 submarine [50], a conceptual vehicle designed to navigate the hydrocarbon seas of Saturn's largest moon, Titan. The use of CFD in a mission of this kind is critical as it can provide important feedback for navigation techniques and ideal locations and depths, and also suggest design modifications to increase the efficiency of the overall mission. Preliminary CFD simulations for this problem can be found in [51, 52], where the submarine is moved at a constant speed of 1 m/s in a large tank made of SPH fluid particles. The open boundary formulation implemented in this work becomes central in simulations of this kind because it allows lower computational time by selecting a smaller numerical domain where the submarine is still and the fluid moves with certain conditions imposed at the domain boundaries.

Figure 19 illustrates the initial configuration with the submarine operating in surfaced conditions. During the mission this is expected to happen around 14 hours per day to allow direct-to-Earth communication. A rectangular fluid domain with dimensions $8 \times 5 \times 1.8$ m^3 is used, while the submarine has a length overall of 6 m and is initially sinked 0.6 m below the free surface. Buffer 1 is the inflow area, with assigned speed $U_\infty = 1$ m/s and density extrapolated from the fluid domain, whereas Buffer 2 is the outflow area with assigned speed $U_\infty = 1$ m/s and assigned density such to retrieve a hydrostatic pressure distribution. In both buffers the free-surface level is extrapolated from the ghost nodes. Particle spacing, fluid properties, Titan's gravitational acceleration, and other simulation parameters are chosen to match those in [51, 52] for comparison of the simulation results. Similar to other cases presented in this work, the ratio of smoothing length to particle spacing is 1.5 and $\delta = 0.1$ for δ-SPH.

Figure 20 shows the normalized free-surface elevation, η/Z, where Z is the submarine total height. A bow wave is generated and expected, with shape and elevation in close agreement with the results obtained in [51, 52]. The deck of the submarine is covered with a layer of liquid resulting from run-off of the bow wave. Due to the relatively low speed of the submarine during navigation at the free surface, part of the liquid is able to rest on top of the submarine at this draft. To better quantify the results, the total hydrodynamic force acting on the hull opposing motion in the forward direction as well as the vertical force are plotted in Figures 21a and 21b, respectively. The hydrodynamic drag in Figure 21a convergences at around 192 N after about 4 seconds of simulated physical time. This value is an excellent match with the 194 N found in [52]. The vertical force in Figure 21b is normalized with respect to the weight of the liquid displaced by the submarine hull, with a value around the unity observed for the majority of the simulation time. The submarine is therefore operating in the displacement hull regime. The downward trend notable towards the final instants of the simulation is likely due to the forcing effect of the liquid accumulating at the bow, thus accounting for the slight drop in fluid buoyant force.

Some quantitative information about the simulation and algorithm efficiency can be found in Table 3. About 13% of the computational time is dedicated to the open boundary treatment, with 250 particles on average generated every time step. The cost needed to enforce open boundary conditions in DualSPHysics is therefore acceptable. Additionally, a 108× speed-up is calculated with respect to adopting the simulation approach in [52], mainly due to having reduced the number of SPH particles by more than one order of magnitude. This proves the importance of open boundary conditions for the solution of this kind of CFD problems.

5. Conclusion

In this paper, a novel methodology for open boundary conditions in SPH has been presented and implemented in the open-source code DualSPHysics. The model is based on the use of buffer layers near the open regions of the computational domain. Particle in these buffers are used as a means of enforcing certain boundary conditions. Specifically, flow variables belonging to buffer particles can be either assigned a priori or extracted from the fluid domain using a first-order accurate ghost-nodes based method. For the velocity and density of the buffer particles, the available options are to impose a given profile, either constant or variable over time, or to interpolate from the fluid domain interior. Moreover, the water depth in the buffer can be imposed to be constant, follow a variable input, or alternatively be extrapolated from the free-surface level in the fluid domain in the vicinity of the buffer.

The algorithm presents many novelties, such as the ability to convey physical information from the fluid domain to the boundary with an accurate and consistent procedure. Additionally, the availability of free-surface extrapolation and time-varying free-surface elevation in the buffer regions allows the simulation of wave generation and other free-surface flows with minimal reflection of numerical noise into the fluid domain.

The proposed algorithm has been tested successfully against a variety of 2-D and 3-D test cases. Results from these simulations corroborate the effectiveness of the presented open boundary formulation in modeling complicated fluid problems, such as three-dimensional flow past a ship hull or wave generation. The algorithm performs well also when coupled with other SPH features, such as particle shifting and δ–SPH.

Future work will be focused on the implementation of hybridization techniques to couple DualSPHysics with other CFD codes using the open boundary algorithm implemented herein. It is expected that the availability of open boundary conditions in a highly parallel open-source SPH code will broaden the use of SPH in a range of engineering problems that are not readily solvable within the current DualSPHysics framework.

References

[1] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society 181 (3) (1977) 375–389. `doi:10.1093/mnras/181.3.375`.

[2] L. B. Lucy, A numerical approach to the testing of the fission hypothesis, Astronomical Journal 82 (12) (1977) 1013–1024. `doi:10.1086/112164`.

[3] J. J. Monaghan, Simulating free surface flows with SPH, Journal of Computational Physics 110 (1994) 399–406. `doi:10.1006/jcph.1994.1034`.

[4] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. L. Touzé, G. Graziani, δ-SPH model for simulating violent impact flows, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1526 – 1542. `doi:10.1016/j.cma.2010.12.016`.

[5] P. L. Tallec, J. Mouro, Fluid structure interaction with large structural displacements, Computer Methods in Applied Mechanics and Engineering 190 (24) (2001) 3039 – 3067, advances in Computational Methods for Fluid-Structure Interaction. `doi:10.1016/S0045-7825(00)00381-9`.

[6] A. Rafiee, K. P. Thiagarajan, An SPH projection method for simulating fluid-hypoelastic structure interaction, Computer Methods in Applied Mechanics and Engineering 198 (33) (2009) 2785 – 2795. `doi:10.1016/j.cma.2009.04.001`.

[7] R. B. Canelas, J. M. Domínguez, A. J. C. Crespo, M. Gómez-Gesteira, R. M. L. Ferreira, A smooth particle hydrodynamics discretization for the modelling of free surface flows and rigid body dynamics, International Journal for Numerical Methods in Fluids 78 (9) (2015) 581–593, fld.4031. `doi:10.1002/fld.4031`.

[8] R. B. Canelas, A. J. C. Crespo, J. M. Domínguez, R. M. L. Ferreira, M. Gómez-Gesteira, SPH–DCDEM model for arbitrary geometries in free surface solid–fluid flows, Computer Physics Communications 202 (2016) 131 – 140. `doi:10.1016/j.cpc.2016.01.006`.

[9] G. Fourtakas, B. Rogers, Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using smoothed particle hydrodynamics (sph) accelerated with a graphics processing unit (gpu), Advances in Water Resources 92 (2016) 186 – 199. `doi:https://doi.org/10.1016/j.advwatres.2016.04.009`.

[10] E. H. Zubeldia, G. Fourtakas, B. D. Rogers, M. M. Farias, Multi-phase sph model for simulation of erosion and scouring by means of the shields and drucker–prager criteria., Advances in Water Resources 117 (2018) 98 – 114. `doi:https://doi.org/10.1016/j.advwatres.2018.04.011`.

[11] A. Khayyer, H. Gotoh, H. Falahaty, Y. Shimizu, Towards development of enhanced fully-lagrangian mesh-free computational methods for fluid-structure interaction, Journal of Hydrodynamics 30 (1) (2018) 49–61. `doi:10.1007/s42241-018-0005-x`.

[12] M. S. Shadloo, G. Oger, D. L. Touzé, Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges, Computers & Fluids 136 (2016) 11 – 34. `doi:10.1016/j.compfluid.2016.05.029`.

[13] D. Violeau, B. D. Rogers, Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future, Journal of Hydraulic Research 54 (1) (2016) 1–26. `doi:10.1080/00221686.2015.1119209`.

[14] C. Altomare, T. Suzuki, J. M. Dominguez, A. Crespo, M. Gomez-Gesteira, I. Caceres, A hybrid numerical model for coastal engineering problems, Coastal Engineering Proceedings 1 (34). `doi:10.9753/icce.v34.waves.60`.

[15] G. Fourey, C. Hermange, D. L. Touzé, G. Oger, An efficient fsi coupling strategy between smoothed particle hydrodynamics and finite element methods, Computer Physics Communications 217 (2017) 66 – 81. `doi:https://doi.org/10.1016/j.cpc.2017.04.005`.

[16] M. Lastiwka, M. Basa, N. J. Quinlan, Permeable and non-reflecting boundary conditions in SPH, International Journal for Numerical Methods in Fluids 61 (7) (2009) 709–724. `doi:10.1002/fld.1971`.

[17] C. E. Alvarado-Rodríguez, J. Klapp, L. D. G. Sigalotti, J. M. Domínguez, E. de la Cruz Sánchez, Nonreflecting outlet boundary conditions for incompressible flows using SPH, Computers & Fluids 159 (2017) 177 – 188. `doi:10.1016/j.compfluid.2017.09.020`.

[18] R. Vacondio, B. D. Rogers, P. K. Stansby, P. Mignosa, Sph modeling of shallow flow with open boundaries for practical flood simulation, Journal of Hydraulic Engineering 138 (6) (2012) 530–541. `doi:10.1061/(ASCE)HY.1943-7900.0000543`.

[19] I. Federico, S. Marrone, A. Colagrossi, F. Aristodemo, M. Antuono, Simulating 2D open-channel flows through an SPH model, European Journal of Mechanics - B/Fluids 34 (2012) 35 – 46. `doi:10.1016/j.euromechflu.2012.02.002`.

[20] M. Ferrand, A. Joly, C. Kassiotis, D. Violeau, A. Leroy, F.-X. Morel, B. D. Rogers, Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D, Computer Physics Communications 210 (2017) 29 – 44. `doi:10.1016/j.cpc.2016.09.009`.

[21] M. Hirschler, P. Kunz, M. Huber, F. Hahn, U. Nieken, Open boundary conditions for ISPH and their application to micro-flow, Journal of Computational Physics 307 (2016) 614 – 633. `doi:10.1016/j.jcp.2015.12.024`.

[22] S. M. Hosseini, J. J. Feng, Pressure boundary conditions for computing incompressible flows with SPH, Journal of Computational Physics 230 (19) (2011) 7473 – 7487. `doi:10.1016/j.jcp.2011.06.013`.

[23] A. Monteleone, M. Monteforte, E. Napoli, Inflow/outflow pressure boundary conditions for smoothed particle hydrodynamics simulations of incompressible flows, Computers & Fluids 159 (2017) 9 – 22. `doi:10.1016/j.compfluid.2017.09.011`.

[24] A. J. C. Crespo, J. M. Domínguez, B. D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, O. García-Feal, DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (sph), Computer Physics Communications 187 (2015) 204 – 216. `doi:10.1016/j.cpc.2014.10.004`.

[25] J. M. Domínguez, A. J. C. Crespo, M. Gómez-Gesteira, Optimization strategies for cpu and gpu implementations of a smoothed particle hydrodynamics method, Computer Physics Communications 184 (3) (2013) 617 – 627. `doi:10.1016/j.cpc.2012.10.015`.

[26] J. M. Domínguez, A. J. C. Crespo, D. Valdez-Balderas, B. D. Rogers, M. Gómez-Gesteira, New multi-gpu implementation for smoothed particle hydrodynamics on heterogeneous clusters, Computer Physics Communications 184 (8) (2013) 1848 – 1860. `doi:10.1016/j.cpc.2013.03.008`.

[27] M. Gómez-Gesteira, A. J. C. Crespo, B. D. Rogers, R. A. Dalrymple, J. M. Dominguez, A. Barreiro, SPHysics – development of a free-surface fluid solver – Part 2: Efficiency and test cases, Computers & Geosciences 48 (2012) 300–307. `doi:10.1016/j.cageo.2012.02.028`.

[28] A. Tafuni, I. Sahin, Non-linear hydrodynamics of thin laminae undergoing large harmonic oscillations in a viscous fluid, Journal of Fluids and Structures 52 (2015) 101 – 117. `doi:10.1016/j.jfluidstructs.2014.10.004`.

[29] A. Tafuni, I. Sahin, M. Hyman, Numerical investigation of wave elevation and bottom pressure generated by a planing hull in finite-depth water, Applied Ocean Research 58 (2016) 281 – 291. `doi:10.1016/j.apor.2016.04.002`.

[30] E. Y. M. Lo, S. Shao, Simulation of near-shore solitary wave mechanics by an incompressible SPH method, Applied Ocean Research 24 (2002) 275–286. `doi:10.1016/S0141-1187(03)00002-6`.

[31] R. A. Dalrymple, B. D. Rogers, Numerical modeling of water waves with the SPH method, Coastal Engineering '53 (2006) 141–147. `doi:10.1016/j.coastaleng.2005.10.004`.

[32] D. Molteni, A. Colagrossi, A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH, Computer Physics Communications 180 (6) (2009) 861 – 872. `doi:10.1016/j.cpc.2008.12.004`.

[33] R. Xu, P. Stansby, D. Laurence, Accuracy and stability in incompressible sph (ISPH) based on the projection method and a new approach, Journal of Computational Physics 228 (18) (2009) 6703 – 6725. `doi:10.1016/j.jcp.2009.05.032`.

[34] R. Vacondio, B. D. Rogers, P. K. Stansby, P. Mignosa, J. Feldman, Variable resolution for SPH: a dynamic particle coalescing and splitting scheme, Computer Methods in Applied Mechanics and Engineering 256 (1) (2013) 132–148. `doi:10.1016/j.cma.2012.12.014`.

[35] R. Vacondio, B. D. Rogers, P. K. Stansby, P. Mignosa, Variable resolution for sph in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity, Computer Methods in Applied Mechanics and Engineering 300 (2016) 442 – 460. `doi:10.1016/j.cma.2015.11.021`.

[36] S. J. Lind, R. Xu, P. K. Stansby, B. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, Journal of Computational Physics 231 (4) (2012) 1499 – 1523. `doi:10.1016/j.jcp.2011.10.027`.

[37] A. Skillen, S. Lind, P. K. Stansby, B. D. Rogers, Incompressible smoothed particle hydrodynamics (SPH) with reduced temporal noise and generalised Fickian smoothing applied to body–water slam and efficient wave–body interaction, Computer Methods in Applied Mechanics

and Engineering 265 (2013) 163 – 173. `doi:10.1016/j.cma.2013.05.017`.

[38] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, P. Stansby, Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method, Journal of Computational Physics 227 (18) (2008) 8417 – 8436. `doi:10.1016/j.jcp.2008.06.005`.

[39] A. Mokos, B. D. Rogers, P. K. Stansby, A multi-phase particle shifting algorithm for sph simulations of violent hydrodynamics with a large number of particles, Journal of Hydraulic Research 55 (2) (2017) 143–162. `doi:10.1080/00221686.2016.1212944`.

[40] A. J. C. Crespo, M. Gómez-Gesteira, R. A. Dalrymple, Boundary conditions generated by dynamic particles in SPH methods, Computers, Materials, & Continua 5 (3) (2007) 173–184. `doi:10.3970/cmc.2007.005.173`.

[41] M. Gómez-Gesteira, B. D. Rogers, A. J. C. Crespo, R. A. Dalrymple, M. Narayanaswamy, J. M. Dominguez, SPHysics – development of a free-surface fluid solver – Part 1: Theory and formulations, Computers & Geosciences 48 (2012) 289–299. `doi:10.1016/j.cageo.2012.02.029`.

[42] S. Marrone, A. Colagrossi, M. Antuono, G. Colicchio, G. Graziani, An accurate SPH modeling of viscous flows around bodies at low and moderate Reynolds numbers, Journal of Computational Physics 245 (2013) 456–475. `doi:10.1016/j.jcp.2013.03.011`.

[43] M. B. Liu, G. R. Liu, Restoring particle consistency in smoothed particle hydrodynamics, Applied Numerical Mathematics 56 (1) (2006) 19 – 36. `doi:10.1016/j.apnum.2005.02.012`.

[44] C. Liu, X. Zheng, C. H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, Journal of Computational Physics 139 (1) (1998) 35–57. `doi:10.1006/jcph.1997.5859`.

[45] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, Journal of Computational Physics 176 (2) (2002) 231–275. `doi:10.1006/jcph.2001.6970`.

[46] R. L. Panton, Boundary layers, John Wiley & Sons, Inc., 2013, pp. 533–606. `doi:10.1002/9781118713075.ch20`.

[47] W. E. Langlois, M. O. Deville, Exact solutions to the equations of viscous flow, Springer International Publishing, Cham, 2014, pp. 105–143. `doi:10.1007/978-3-319-03835-3_4`.

[48] C. Altomare, J. M. Domínguez, A. J. C. Crespo, J. González-Cao, T. Suzuki, M. Gómez-Gesteira, P. Troch, Long-crested wave generation and absorption for SPH-based dualSPHysics model, Coastal Engineering 127 (2017) 37 – 54. `doi:10.1016/j.coastaleng.2017.06.004`.

[49] C. Altomare, J. M. Domínguez, A. J. C. Crespo, T. Suzuki, I. Caceres, M. Gómez-Gesteira, Hybridization of the wave propagation model swash and the meshfree particle method sph for real coastal applications, Coastal Engineering Journal 57 (4) (2015) 1550024–1–1550024–34. `doi:10.1142/S0578563415500242`.

[50] S. R. Oleson, R. D. Lorenz, M. V. Paul, Phase 1 final report: Titan submarine, Tech. rep., NASA Glenn Research Center; Cleveland, OH United States (2015).

[51] S. Carberry Mogan, P. Sawicki, C. J. Bernardo, D. Chen, I. Sahin, J. Hartwig, A. Tafuni, CFD study of an autonomous submarine in extraterrestrial seas, in: Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE, Volume 1, Cleveland, OH, August 6–9, 2017. `doi:10.1115/DETC2017-67593`.

[52] S. Carberry Mogan, D. Chen, C. J. Bernardo, I. Sahin, J. Hartwig, S. Oleson, A. Tafuni, Numerical simulations of flow past a submarine in extraterrestrial, cryogenic seas, in: Proceedings of the AIAA SPACE and Astronautics Forum and Exposition, AIAA SPACE Forum, (AIAA 2017-5204), Orlando, FL, September 12–14, 2017. `doi:110.2514/6.2017-5204`.

Table 1: Values of the Strouhal number for the unsteady cases.

	SPH (Present)	Liu et al. (1998) [44]	Calhoun (2002) [45]
Re = 100	0.177	0.165	0.175
Re = 200	0.206	0.192	0.202
	SPH (Present)	Marrone et al. (2013) [42]	Vacondio et al. (2013) [34]
Re = 100	0.177	0.168	0.175
Re = 200	0.206	0.200	-

Table 2: Values of the lift coefficient, C_L, for the unsteady cases.

	SPH (Present)	Liu et al. (1998) [44]	Calhoun (2002) [45]
Re = 100	0.322	0.339	0.298
Re = 200	0.693	0.690	0.668
	SPH (Present)	Marrone et al. (2013) [42]	Vacondio et al. (2013) [34]
Re = 100	0.322	0.240	0.330
Re = 200	0.693	0.680	-

Table 3: Simulation specifications on a NVIDIA Tesla K80 with 24GB GDDR5.

	Without OBC [52]	With OBC
Domain Size	$42 \times 21 \times 4$ m^3	$8 \times 5 \times 1.8$ m^3
# of particles	6.7×10^7	1.4×10^6
Calculated force	194.29 [N]	191.8 [N]
Simulated physical time	12 [s]	12 [s]
Simulation total time	378.90 [h]	3.5 [h]
Speed-Up	–	108x
GPU memory requirement	~ 6 GB	~ 0.7 GB

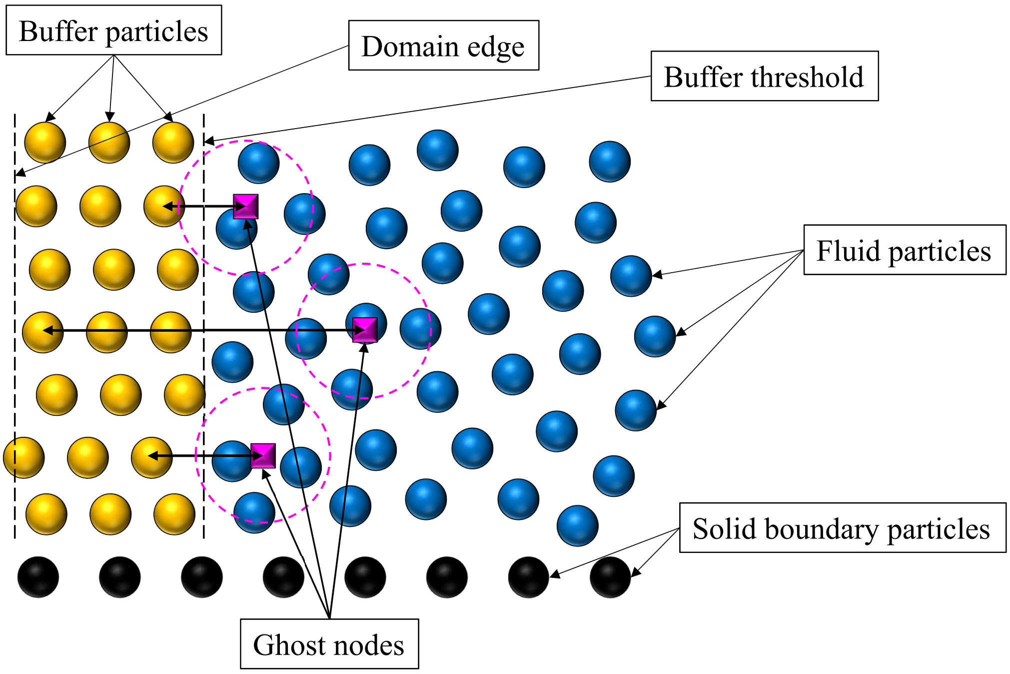
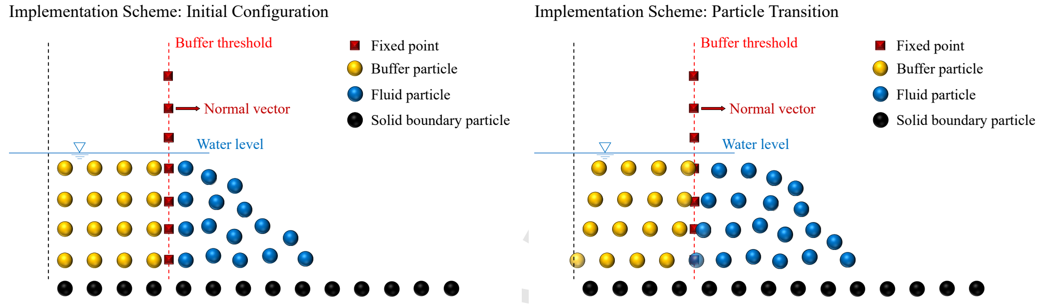
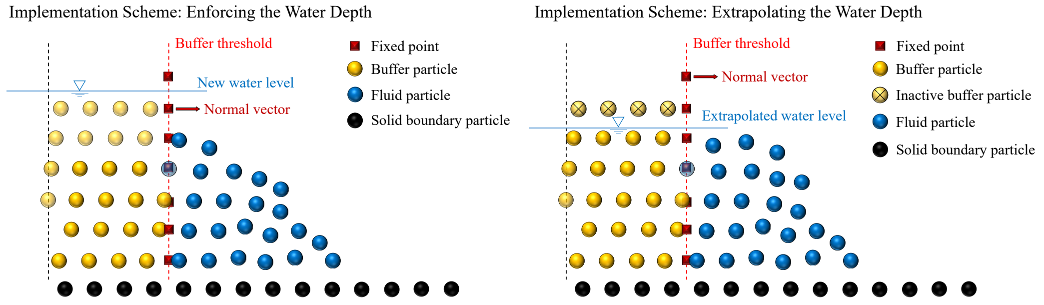


Figure 1: Sketch of the implemented open boundary formulation.



(a) Buffer area in its initial flow configuration.

(b) Buffer evolution with unchanged water depth.



(c) Buffer evolution with variable water depth externally enforced.

(d) Buffer evolution with variable water depth obtained from the fluid domain.

Figure 2: Sketch of the operating principles of the proposed algorithm for different free-surface flow cases.

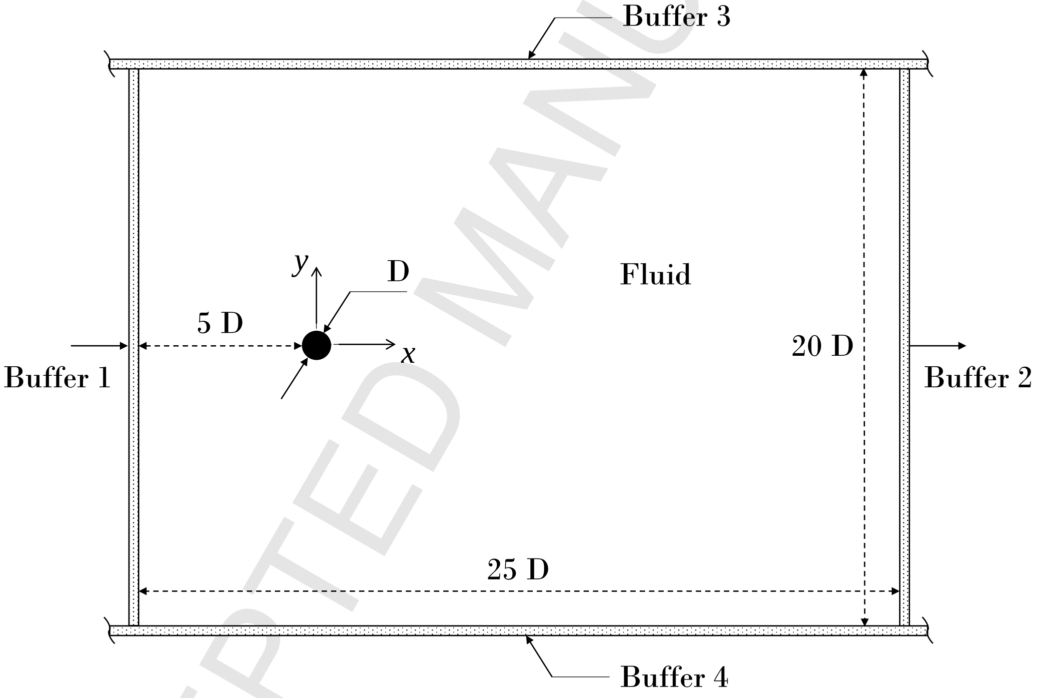
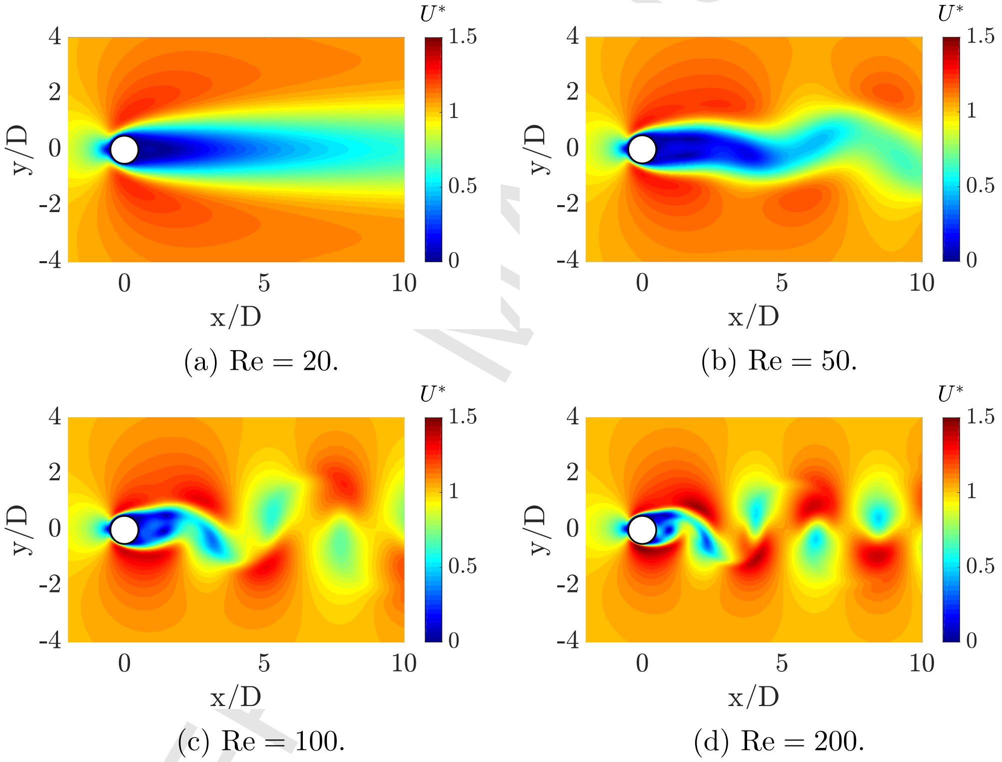


Figure 3: Computational domain for 2-D flow past a circular cylinder.



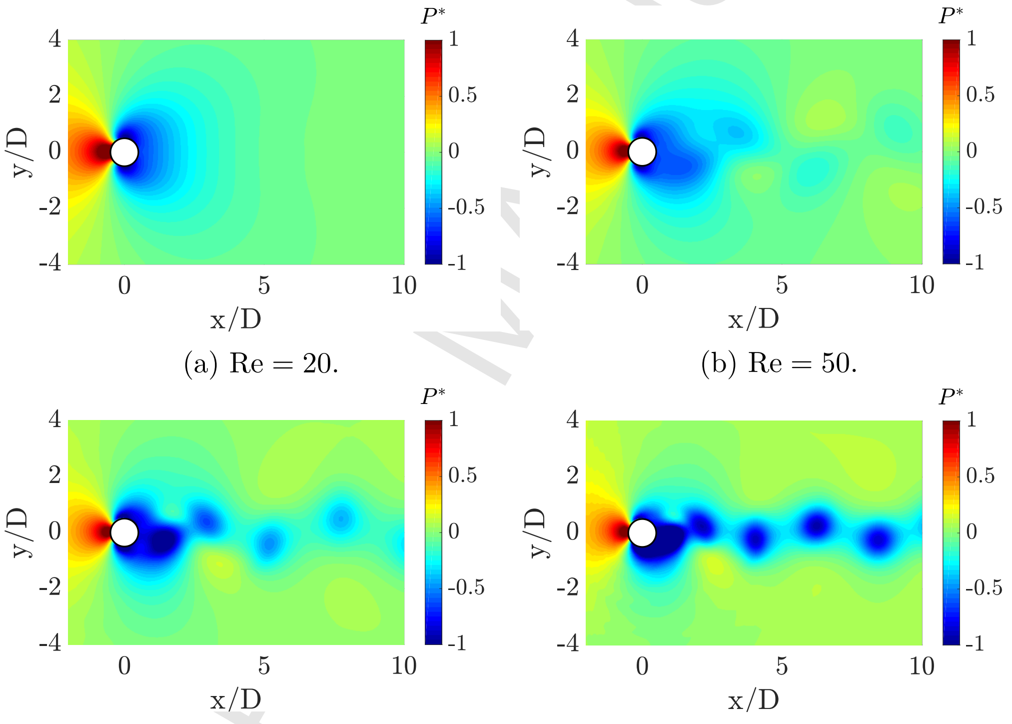
(a) Re = 20.

(b) Re = 50.

(c) Re = 100.

(d) Re = 200.

Figure 4: 2-D flow past a cylinder: velocity contours.



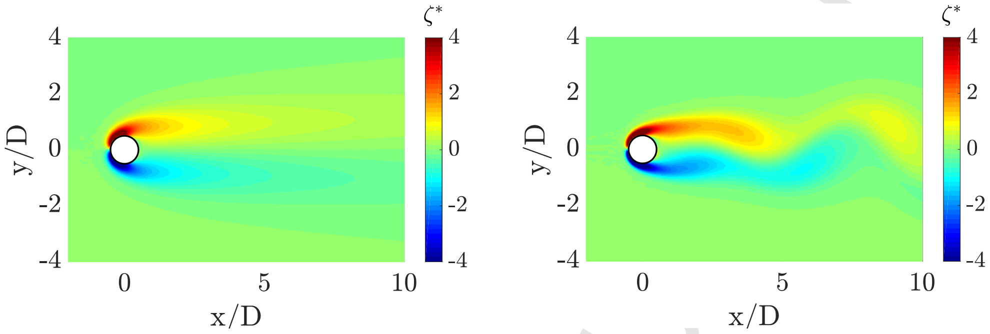
(a) Re = 20.

(b) Re = 50.

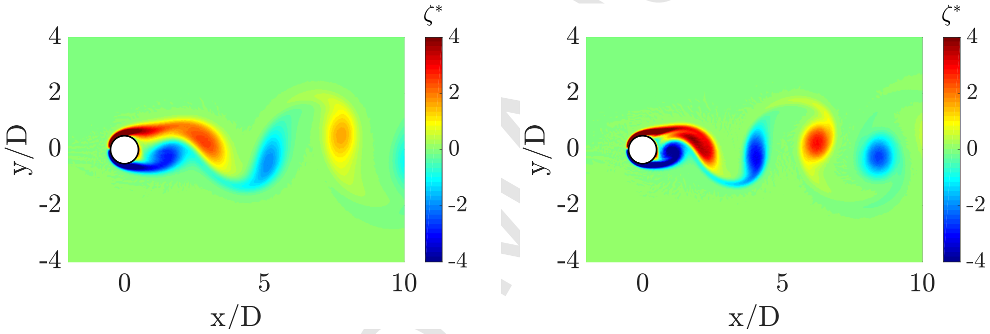
(c) Re = 100.

(d) Re = 200.

Figure 5: 2-D flow past a cylinder: pressure contours.

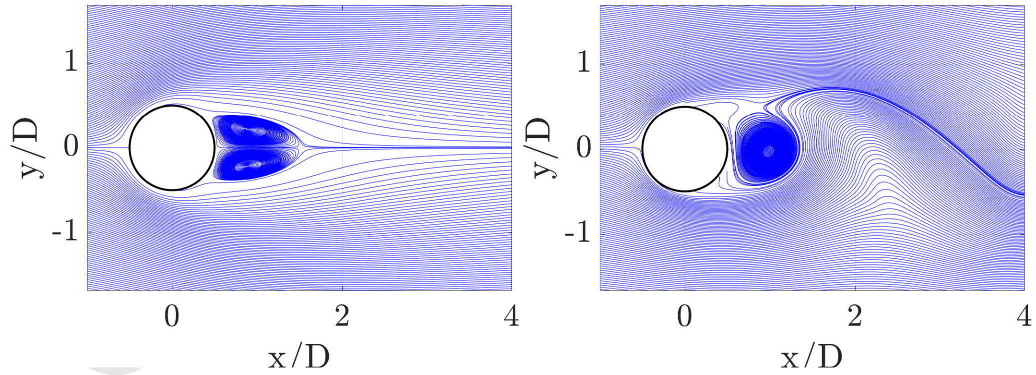


(a) Re = 20. (b) Re = 50.



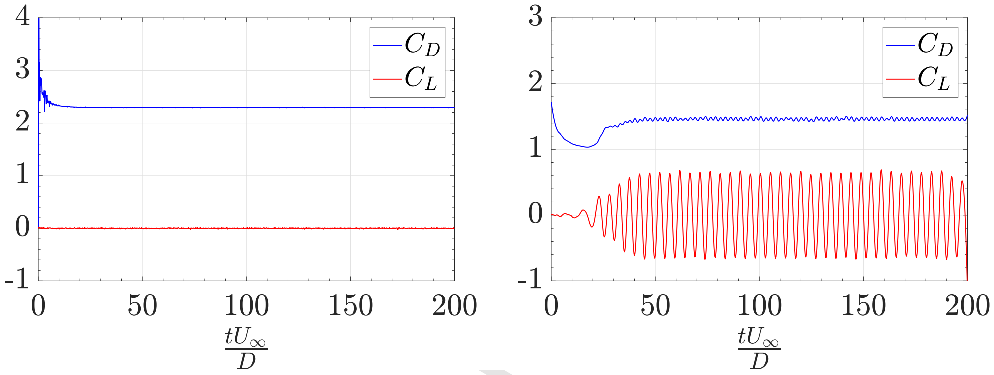
(c) Re = 100. (d) Re = 200.

Figure 6: 2-D flow past a cylinder: vorticity contours.



(a) Re = 20. (b) Re = 200.

Figure 7: 2-D flow past a cylinder: streamlines.



(a) Re = 20.

(b) Re = 200.

Figure 8: Time history of drag and lift coefficients for flow past a circular cylinder.

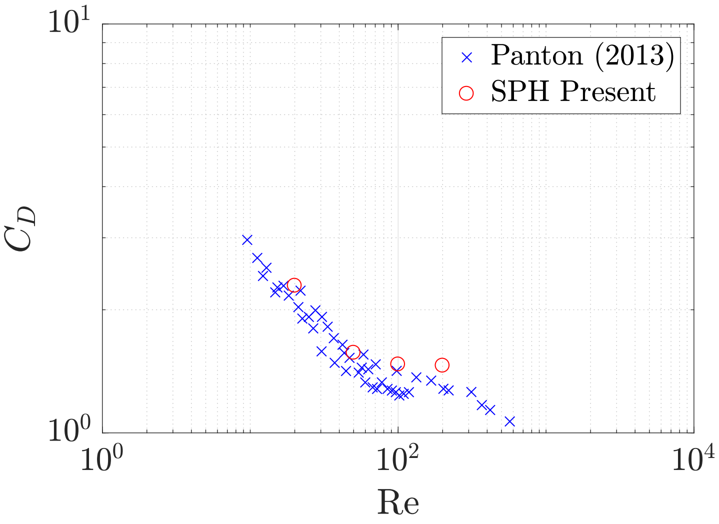


Figure 9: Drag coefficient as a function of the Reynolds number for flow past a circular cylinder.

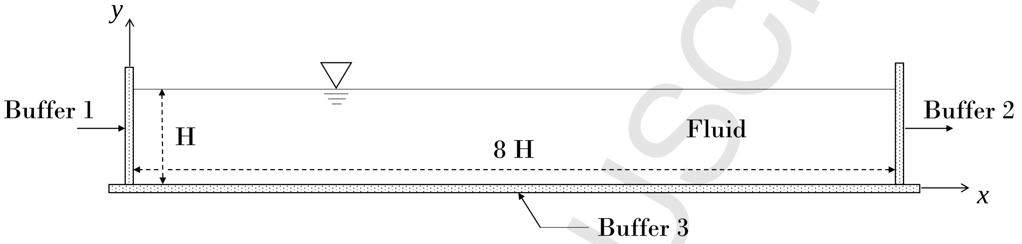


Figure 10: Computational domain for 2-D open-channel flow.

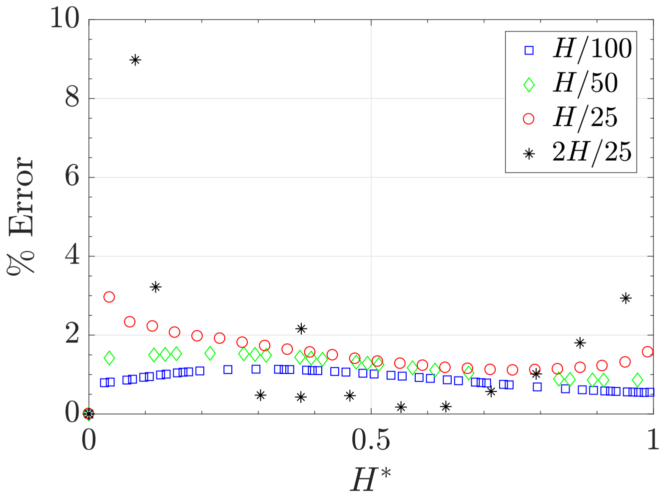
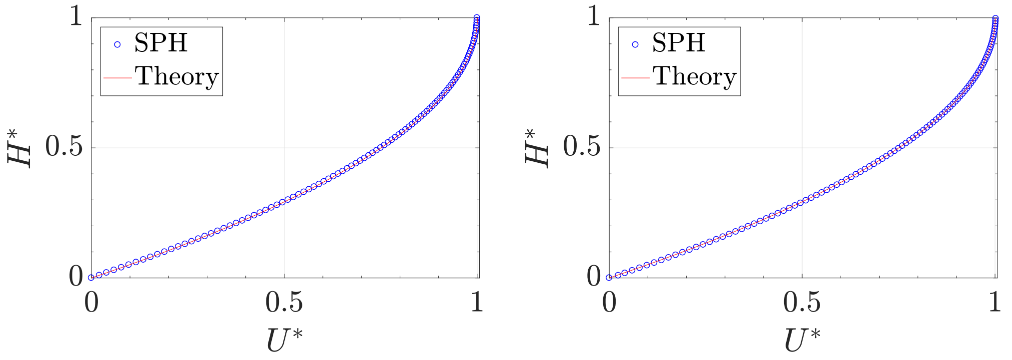
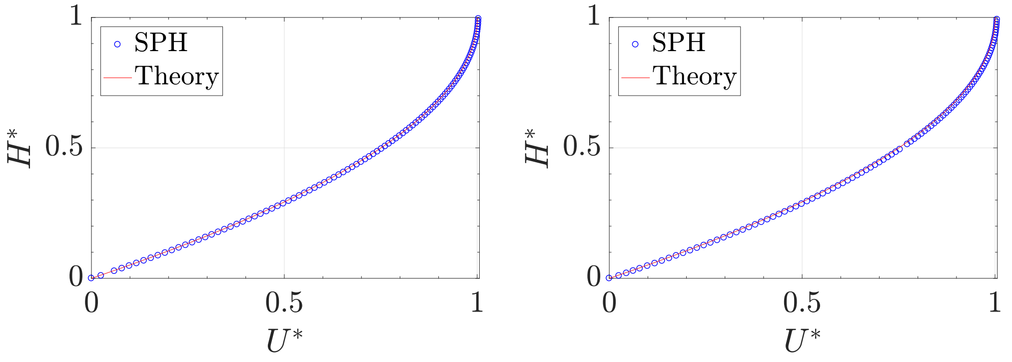


Figure 11: Percent error calculated against the analytical open-channel flow velocity for different particle resolutions at time $t = 20$ s.



(a) Inlet, $\frac{x}{L} = 0$.   (b) $\frac{x}{L} = \frac{1}{3}$.



(c) $\frac{x}{L} = \frac{2}{3}$.   (d) Outlet, $\frac{x}{L} = 1$.

Figure 12: Analytical (solid line) and SPH (circles) open-channel flow velocity profiles at different x locations at time 20 s for simulation with resolution $H/100$.

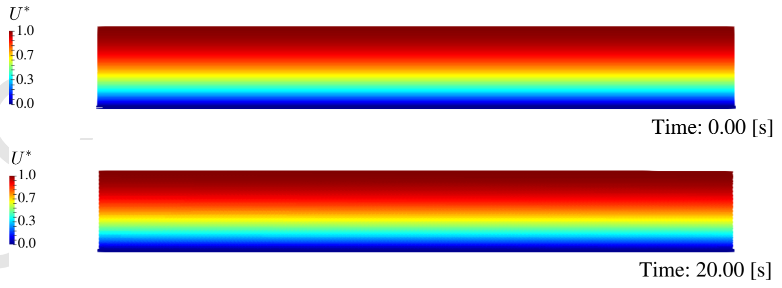


Figure 13: Contours of the velocity magnitude for the open channel flow with resolution $H/100$: analytical solution (top) and numerical results at time 20 s (bottom) .

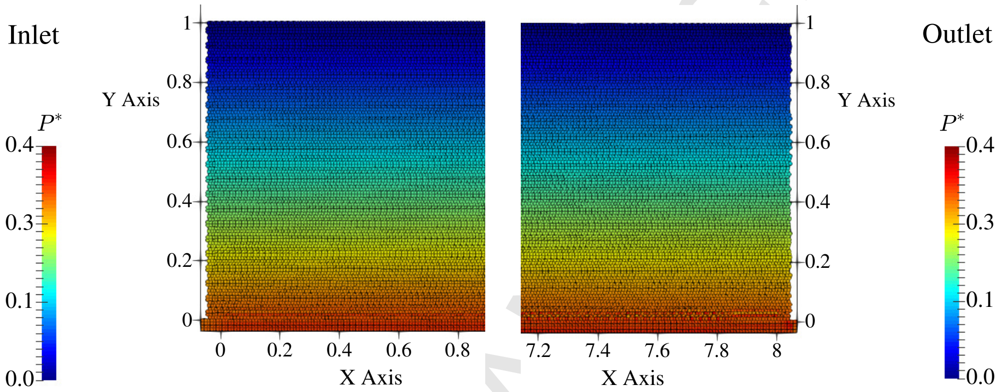


Figure 14: Close-up of pressure contours for open-channel flow at $t = 20$ s with resolution $H/100$.

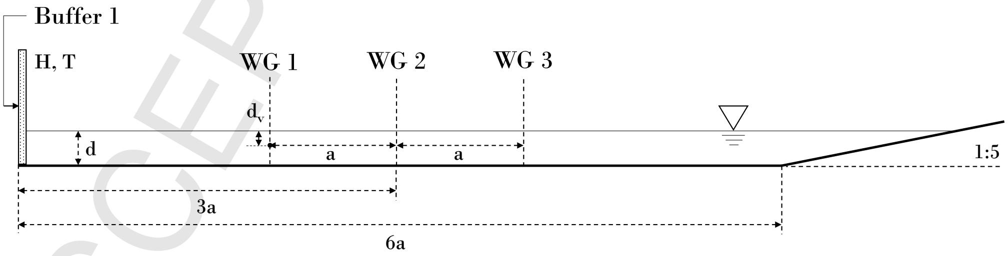


Figure 15: Computational domain for the 2-D wave tank.

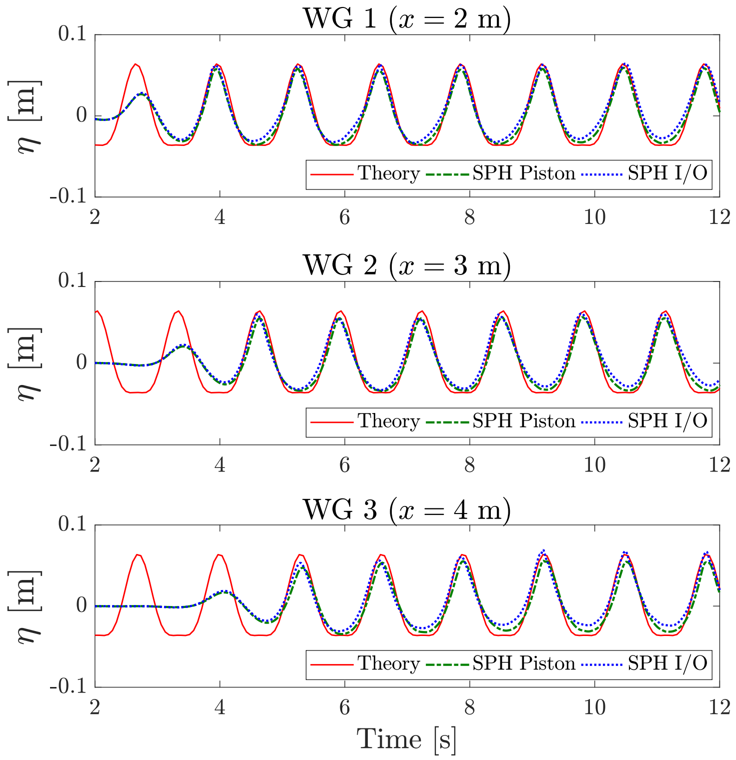


Figure 16: Comparison of theoretical and numerical free surface elevations at different wave gauges for the wave generation study.

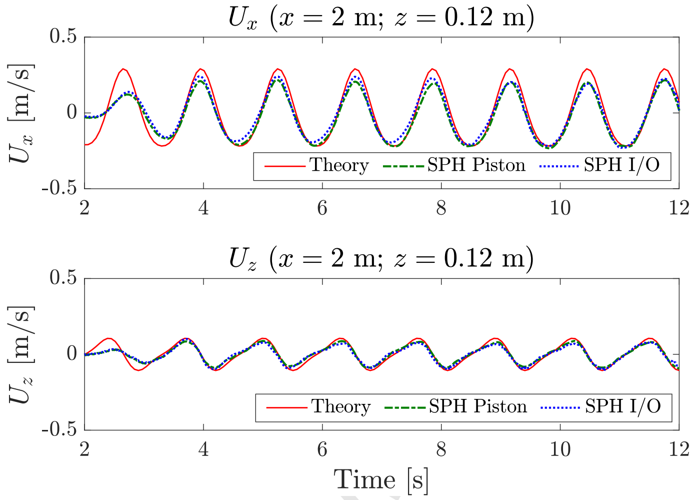


Figure 17: Comparison of theoretical and numerical orbital velocities at the velocity gauge for the wave generation study.

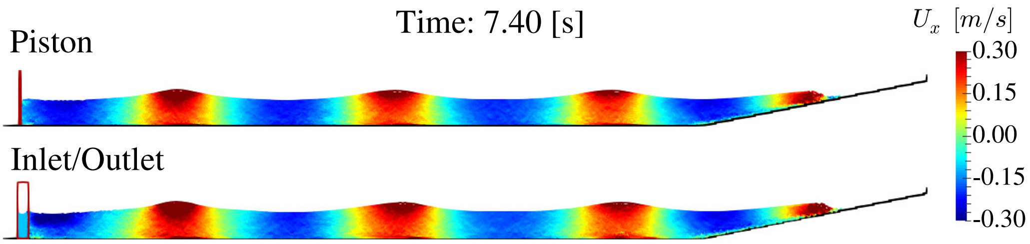


Figure 18: Snapshot of the simulation with piston and I/O conditions at $t = 7.40$ s for the wave generation study.

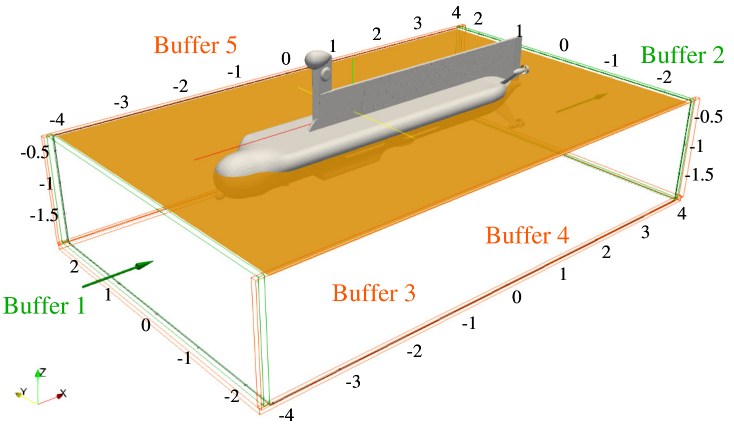


Figure 19: Simulation set-up for flow past the Titan submarine.

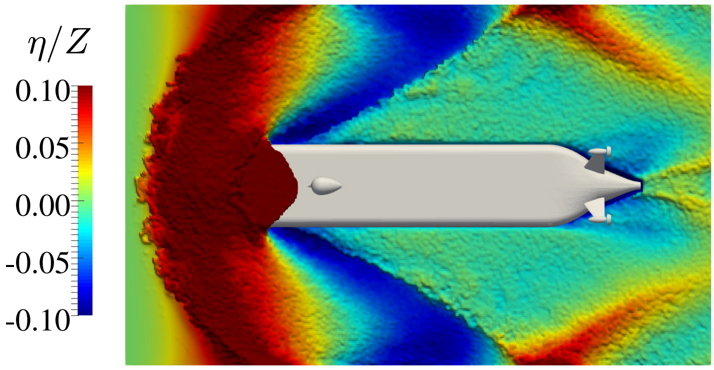
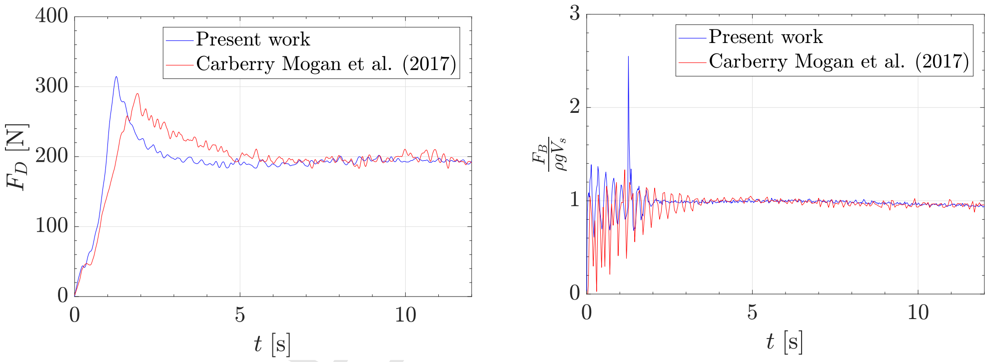


Figure 20: Contours of free-surface elevation around the Titan submarine.



(a) Drag force, F_D.

(b) Normalized buoyant force, $\frac{F_V}{\rho g V_s}$.

Figure 21: Time history of fluid forces on the Titan submarine.