Boundary condition enforcement for renormalised weakly compressible meshless Lagrangian methods

Johannes Joubert, Daniel Wilke, Nicolin Govender, Patrick Pizette, Josip Basic, Nor-Edine Abriak

Version of Record: https://www.sciencedirect.com/science/article/pii/S0955799721001156
Manuscript_02f809b9c76f287f7b3e7c225ea168aa

Boundary condition enforcement for renormalised weakly compressible meshless Lagrangian methods

Johannes C. Joubert[a,b,*], Daniel N. Wilke[a], Nicolin Govender[c,d], Patrick Pizette[b], Josip Basic[e], Nor-Edine Abriak[b]

[a] *Department of Mechanical and Aeronautical Engineering, University of Pretoria, South Africa*

[b] *IMT Lille Douai, Univ. Lille, EA 4515 - LGCgE Laboratoire de Génie Civil et géoEnvironnement, CERI Matériaux et Procédés, F-59000 Lille, France*

[c] *University of Johannesburg, South Africa*

[d] *RCPE, Graz, Austria*

[e] *Department of Naval Architecture, University of Split, Rudera Boskovica 32, 21000 Split, Croatia*

Abstract

This paper introduces a boundary condition scheme for weakly compressible (WC) renormalised first-order accurate meshless Lagrangian methods (MLM) by considering both solid and free surface conditions.

A hybrid meshless Lagrangian method-finite difference (MLM-FD) scheme on prescribed boundary nodes is proposed to enforce Neumann boundary conditions. This is used to enforce symmetry boundary conditions and the implied Neumann pressure boundary conditions on solid boundaries in a manner consistent with the Navier-Stokes equation leading to the accurate recovery of surface pressures. The free surface boundary conditions allow all differential operators to be approximated by the same renormalised scheme while also efficiently determining free surface particles.

The boundary conditions schemes are implemented for two renormalised MLMs. A WC smoothed particle hydrodynamics (SPH) solver is compared to a WC generalised finite difference (GFD) solver. Applications in both 2D and 3D are explored. A substantial performance benefit was found when comparing the WCGFD solver to the WCSPH solver with the WCGFD solver realising a maximum speedup in the range of three times over WCSPH in both 2D and 3D configurations. The solvers were implemented in C++ and used the NVIDIA CUDA 10.1 toolkit for the parallelisation of the solvers.

Keywords: Smoothed Particle Hydrodynamics (SPH), General Finite Difference (GFD), Meshless Lagrangian Method (MLM), Weakly compressible, Boundary conditions, Free surface

1. Introduction

The rapid development of general-purpose graphics processing unit (GPGPU) computing has led to various engineering problems to be recast into a parallelised form with benefits in terms of computational cost and efficiency being shown in both cases. In a computational fluid dynamics (CFD) setting, one way in which this is being realised is through the development of the so-called meshless Lagrangian methods (MLMs). In these systems, clouds of sampling points with no specifically enforced structure between each other are used to evaluate an underlying field representing the solution of partial differential equations (PDEs). Interpolation techniques are then used to build approximations of these fields and the differential operators required

*Corresponding author
Email address: johannes.joubert@imt-lille-douai.fr (Johannes C. Joubert)

to evolve the system. Due to the meshless nature of these methods, it has been seen that multi-phase [1–4], multi-species [5–7] and coupled multi-physics [8–13] applications can be handled naturally.

Over the years, due to its inherently independent nature, a wide range of applications and relative ease of implementing, smoothed particle hydrodynamics (SPH) has benefited from a tremendous amount of interest leading to SPH almost becoming synonymous with MLMs. Built upon the idea of constructing smooth functions from discrete points and applying differential operators directly to this field, these differential operators are used to deconstruct the Navier-Stokes equation (NSE) into a set of ordinary differential equations.

When applied to a general particle distribution, the original SPH differential operators generate zeroth-order accurate discretisations. This was improved when Randles and Libersky [14] introduced a correction to the SPH gradient operator obtained by truncating the leading error term of the discretisation resulting in a first-order accurate gradient discretisation. Fatehi and Manzari [15] later extended this and obtained a first-order accurate Laplacian and thus allowed for first-order approximations of all terms in the incompressible NSE. A linear system for both gradients and Laplacians must be constructed and inverted for each particle to apply these techniques. Renormalisation can drastically improve the overall results of SPH despite relaxing the conservation of momentum constraint imposed by the classical SPH differential operators [16], while also reducing the kernel support size requiring fewer particles to construct the differential operators [17, 18]. The smaller support region and higher quality discretisations, especially in regions with reduced support, has led to these techniques starting to see wide adoption in modern SPH simulations with various studies either using only gradient correction [19–23] or fully first-order accurate operators [11, 12, 24, 25]. With the renormalised Laplacian operator only being developed recently [15], many of the studies implementing only gradient correction never had the option to include the Laplacian correction. However, several more recent studies have also shied away from the Laplacian correction due to its complexity and high computational cost. This is attributed to the construction of the renormalisation operator which requires the assembly of fourth-order tensors (although, due to symmetry in its entries, this system reduces to a $(d^2 + d)/2$ linear system in d spatial dimension) as well as increasing the number of linear operations required in the construction of the Laplacian itself further increasing its computational cost.

Although first being described in the context of SPH, renormalisation techniques have been applied to other MLMs. The moving particle semi-implicit (MPS) method [26], proposed initially as a semi-implicit method for incompressible fluid simulations and later extended to a weakly compressible (WC) context [27–30], constructs its differential operators by weighting individual finite difference (FD) terms between particles. While conceptually different to the SPH approximations in the sense that differential operators are never applied to continuous field approximations but rather discretise the differential operators themselves, functionally, both schemes suffer from similar issues. Specifically, both schemes are zeroth-order accurate, leading to poor approximations in regions with reduced support. This has led to renormalisation methods applied to MPS differential operators [31–33] yielding similar benefits compared to the SPH case.

Alternatively, the generalised finite difference (GFD) method [34–36] builds gradient and Laplacian operators. These operators are equivalent to the above mentioned fully first-order renormalised operators by directly using Taylor series expansions between a particle and those in its neighbourhood along with kernel weighting functions to construct a weighted least-squares problem. Equivalently, this can be considered as approximating the function gradient by the slope of a plane constructed by best-fitting it to discrete function values. Conceptually, this is unique from other renormalised meshless schemes where usually a lower-order approximation is first constructed and then its leading error term is truncated. This approach has been used to simulate incompressible viscous flow with a fully implicit scheme [37].

Boundary conditions have proven to be challenging for MLMs due to the particles' Lagrangian nature implying that interpolation points near boundaries are generally advected away from their initial positions. This, along with reduced kernel support at the boundaries introducing error in the MLM approximations, leads to several challenges ranging from accurately enforcing Neumann boundary conditions to more fundamental challenges such as ensuring fluid particles remain in the prescribed domain. Despite these challenges, various methods for implementing solid boundaries exist with solutions such as fictitious particles, repulsive boundary particles and semi-analytical boundary conditions having been proposed.

While prescribed fictitious particles have been shown capable of successfully enforcing boundary conditions for solid surfaces [38–41], the work of Violeau and Rogers [42] highlighted that these methods tend to suffer from non-physical separation of fluid particles from the boundary, fluid particles penetrating the boundary and artificial boundary layer formation. An alternative proposed by Libersky and Petschek [43] is to dynamically mirror fluid particles across nearby boundaries and link particle

information between the two sets. This method's effectiveness and relative simplicity for implementing either full- or no-slip boundary conditions have led to mirrored boundary particles becoming very common in SPH simulations. However, despite handling simple geometries very well, mirroring techniques become challenging to implement when complex boundaries are considered. Modern mirroring techniques overcome this limitation by either using prescribed internal-mirror particle pairs where the virtual internal particles obtain information via interpolation of the fluid fields [21, 22, 44, 45] or with locally mirrored particles where each particle creates a collection of mirrored particles that only it sees [46–48]. Marrone et al. [21] further applied a mirroring technique on mirrored particles themselves allowing for regions inaccessible via the one-step mirroring process to be meshed to handle sharp corners. However, despite the progress already made, geometries can still be constructed where mirroring techniques struggle to generate ghost particles unambiguously or fully complete the kernel support. Although solutions can be proposed for specific cases, implementing the schemes becomes challenging due to the number of failure methods that must be addressed individually, especially in 3D simulations.

Repulsive boundary particles introduce artificial forcing conditions at boundaries to prevent SPH particles from escaping the prescribed fluid domain. Proposed initially by Monaghan [49], the distance between SPH and boundary particles are used to prescribe a force on the SPH particles based on the Lennard–Jones potential function. This was later refined such that particles flowing parallel to the boundary would receive a steady forcing condition [50]. An improved boundary forcing condition was introduced by Monaghan and Kajtar [51]. Here, the sum of the discrete forces is independent of the boundary particle spacing ensuring a smooth variation of the boundary force. Although these methods prevent fluid particles from escaping the prescribed fluid domain, the SPH approximations near boundaries remain poor due to the reduced kernel support not being addressed. This reduces the overall accuracy of solutions employing repulsive boundary particles [52].

The semi-analytical boundary method first proposed by Kulasegaram et al. [53] truly accounts for the reduced kernel support near boundaries by directly introducing surface effects into the SPH approximations. The modified approximation reduces to the standard SPH approximation when evaluated far from a boundary. A key challenge of this method has been to discretise the surface integral appropriately. In 2D, this can be handled elegantly since one only needs to consider intersections between line segments and the support domain. However, the 3D case requires building the surface integral over various generalised 2D sections that must further be deconstructed into primitive geometries [54]. This drastically increases the complexity of a general 3D simulation. Although the method is relatively new compared to other SPH boundary condition methods, it applies to many applications [52, 55–59]. Furthermore, while repulsive and fictitious particles offer suitable methods for enforcing full- or no-slip boundary conditions, Ferrand et al. [52] and Mayrhofer et al. [56] extended the method to allow generalised boundary conditions to be enforced as well.

While it is most common for variations of the above methods to enforce boundary conditions, an alternative collection of methods has been proposed where function values of boundary particles are assigned directly such that the required Neumann boundary conditions are satisfied upon building the discretised gradient. The work of Hashemi et al. [17] and Joubert et al. [25] made use of the NSE discretised by the renormalised SPH operators at boundary particles to generate a system of equations describing the surface pressure in a way that incorporates the dynamics of both solid and fluid phases. With the surface pressure naturally obtained from a momentum conservation argument, this information also served as an effective way of coupling the solid and fluid phases. A similar idea was proposed by Basic et al. [37] where GFD operators were used to discretise the system. Here, to further improve the quality of the obtained surface pressure, fluid particles near the solid boundaries were first projected onto the solid surface. A hybrid MLM-FD gradient was then used to extrapolate the pressure field such that the NSE is satisfied on the boundary.

WCSPH implicitly handles free surface boundary conditions. This can be understood by viewing the lacking information in areas with reduced kernel support as equivalent to an approximation of the zero pressure Dirichlet condition. Since renormalisation extrapolates information into under-supported regions, the natural enforcement of free surface conditions is lost with first-order meshless methods. While incompressible SPH (ISPH) handles this by enforcing surface pressure boundary conditions in the pressure Poisson equation directly [22], less elegant solutions have been proposed for WCSPH. Specifically, in the work of Oger et al. [16], the proposed method uses first-order gradients for fully supported particles while alternative operators update free surface particles. Regardless of the mass-momentum coupling used, both schemes require identifying the free surfaces which can be computationally expensive to determine.

A new free surface boundary scheme is presented to handle free surface systems for the first-order WC model. Additionally,

the hybrid MLM-FD based solid boundary condition scheme of [37] is extended to make use of a static mesh allowing one to bypass the projected boundary particle generation step.

Due to first-order SPH operators requiring the construction and inversion of two linear systems for gradient and Laplacian approximations, the complexity and computational cost are drastically increased compared to the standard SPH methodology. This raises the question of whether it is still sensible to consider the original SPH field approximation as the foundation for the formulation of first-order accurate MLMs.

This paper investigates this by comparing the proposed WCGFD method to a similar first-order accurate WCSPH method. A detailed comparison between the computational time and numerical results between both solvers is performed with applications to both submerged and free surface systems being explored.

Following this, the contributions of this paper include:

- The proposal of a free surface boundary condition enforcement scheme in the context of first-order WC MLMs,
- Extending the MLM-FD hybrid scheme for Neumann boundary conditions in the context of CFD to a general boundary mesh avoiding the projection step,
- Proposing the WCGFD scheme to improve the computational cost of first-order WCSPH models.

2. Numerical methods

With this paper's focus around boundary condition enforcement in the context of incompressible CFD, traditionally the incompressible NSE would be discretised by the proposed methods. For weakly compressible MLMs, however, a more general system of partial differential equations (PDEs) is proposed:

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u}, \tag{1}$$

$$\frac{d\mathbf{u}}{dt} = -\frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{u} + \mathbf{g}. \tag{2}$$

Here, (1) is the generalised mass continuity equation while (2) is the incompressible momentum balance of the fluid with ρ the fluid density, $\mathbf{u}$ the fluid velocity, p the pressure, ν the kinematic viscosity and $\mathbf{g}$ the acceleration due to body forces.

It should be noted that while the intent is to simulate incompressible flow, (1) is kept as the generalised continuity equation. This is because applying the WC formulation to couple the mass and momentum equations requires the density to evolve.

Of course, since MLMs are based upon PDEs in their Lagrangian formulation, both (1) and (2) contain total time derivatives of their respective primitive variables.

2.1. Equations of state

As opposed to fully incompressible methods that obtain system pressures through projection methods [20, 60], the WC formulation directly couples the mass and momentum equation with a stiff equation of state. First described by Cole and Weller [61] in the context of underwater explosions, the equation of state used in this study relates density to the pressures by:

$$p_i = \frac{\rho_0 c_s^2}{\gamma}\left(\left(\frac{\rho_i}{\rho_0}\right)^\gamma - 1\right), \tag{3}$$

where ρ_0 is the reference density, γ is the coefficient of expansion usually taken to be 7 [49] and c_s is the speed of sound. It should be emphasised that this speed of sound is not a physical parameter of the system, but rather describes the speed at which information can propagate through the system in a numerical sense while also determining the stiffness of (3). Then c_s is a tuning parameter that controls the trade-off between the allowable time-step size of the numerical integration scheme and the system's incompressibility.

To ensure that the model represents an incompressible system, it has been suggested that the lower limit on c_s should be at least 10 times the maximum expected fluid velocity [49].

2.2. SPH formulation

The SPH formulation, first presented in 1977 by Gingold and Monaghan [62] in the context of astronomical gas system simulations, relies on the building of a smooth function f via an approximation of the following identity:

$$f(\mathbf{r}_i) = \int_{\mathbb{R}^d} f(\mathbf{r})\delta(\mathbf{r} - \mathbf{r}_i)dV, \tag{4}$$

using a compact kernel basis $W(\mathbf{r} - \mathbf{r}_i, h)$ with the key property that $W(\mathbf{r} - \mathbf{r}_i, h) \to \delta(\mathbf{r} - \mathbf{r}_i)$ as $h \to 0$. This paper uses a quintic kernel:

$$W(\mathbf{r}, h) = \sigma \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5 & \text{for } 0 \le q \le 1 \\ (3-q)^5 - 6(2-q)^5 & \text{for } 1 < q \le 2 \\ (3-q)^5 & \text{for } 2 < q \le 3 \\ 0 & \text{otherwise} \end{cases} \quad \text{with} \quad q = \|\mathbf{r}\|_2/h, \tag{5}$$

where σ is the kernel normalisation parameter [39]. If, however, gradient and Laplacian renormalisation are considered, σ plays no role other than scaling the renormalisation tensors and so is set to 1 in this work.

A quintic kernel is chosen over lower-order alternatives due to the discretisation error of the differential operators being related to the smoothness of the kernel function. Specifically, the discretisation error for both SPH gradients and Laplacians used in this work was shown to be proportional to $\left(\delta_0/h\right)^\beta$ [15] where δ_0 is the initial particle spacing and β is the largest integer such that all derivative up to order β are zero on the boundary. Although not offering a benefit in terms of convergence for a fixed δ_0/h, the quintic kernel is expected to improve upon the discretisation error compared to lower-order kernels. With an increase in δ_0/h indicating a decrease in the number of neighbouring particles, this effect does diminish with shrinking kernel support. Although outside the scope of this work, this presents an opportunity to investigate the MLM discretisations' behaviour as a function of number of neighbours and kernel order.

Combining this with a collection of local points indexed by the set $I = \{j \in \mathbb{Z}/N\mathbb{Z} | W(\mathbf{r}_j - \mathbf{r}_i, h) > 0\}$, where N is the total number of points, allows a field to be approximated by a weighted interpolation strategy:

$$f(\mathbf{r}_i) \approx \langle f(\mathbf{r}_i)\rangle = \langle f\rangle_i = \sum_{j\in I} V(\mathbf{r}_j) f(\mathbf{r}_j) W(\mathbf{r}_i - \mathbf{r}_j, h) = \sum_{j\in I} V_j f_j W_{ij}, \tag{6}$$

where f_j represent the function value at the point $\mathbf{r}_j$, V_j represents the volume associated to the "fluid particle" at $\mathbf{r}_j$ which has a relation to the fluid density in the original formulation and $\langle f\rangle_i$ being used to denote the kernel approximation of a function f at $\mathbf{r}_i$.

A key observation to be made is that the approximation $\langle f\rangle_i$ is now also a smooth function of $\mathbf{r}_i$ and as such can be used to determine gradient information directly. With some manipulation of $\langle cf\rangle_i$, where c is a constant, it can be shown that the gradient of f can be approximated as:

$$\langle \nabla f\rangle_i = \sum_{j\in I} V_j (f_j - f_i) \nabla W_{ij}. \tag{7}$$

The Laplacian approximation is not handled in the same fashion as it has been shown that second derivatives of (6) can lead to inconsistent results for the case of unordered particle distributions [63]. Instead, the Laplacian is approximated with a hybrid SPH-FD scheme proposed by Morris et al. [39]. The original expression is modified to capture the full Hessian tensor which is given by:

$$\langle \nabla \otimes \nabla f\rangle_i = 2 \sum_{j\in I, i\neq j} V_j \left(\frac{f_i - f_j}{r_{ij}^2}\right) \mathbf{r}_{ij} \otimes \nabla W_{ij}, \tag{8}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = \|\mathbf{r}_{ij}\|_2$ and $\otimes$ indicates the outer product.

To improve the gradient operator's discrete approximation, Randles and Libersky [14] proposed a new gradient scheme modifying (7) to ensure first-order accuracy. It was shown that it becomes possible to truncate the leading error term of (7) by applying a linear mapping to the gradient approximation:

$$\langle \nabla f \rangle_i = \mathbf{B}_i \cdot \sum_{j \in I} V_j (f_j - f_i) \nabla W_{ij}, \tag{9}$$

where the truncation tensor is given as:

$$\mathbf{B}_i = -\left(\sum_{j \in I} V_j \mathbf{r}_{ij} \otimes \nabla W_{ij} \right)^{-1}, \tag{10}$$

Fatehi and Manzari [15] derived the same form as (9) using a Taylor series expansion of (7). Furthermore, they applied similar techniques to (8) to obtain a first-order accurate Laplacian approximation. The corrected Laplacian approximation is given as:

$$\left\langle \nabla^2 f \right\rangle_i = 2 \sum_{j \in I, i \neq j} V_j \left(\frac{f_i - f_j}{r_{ij}} - \mathbf{e}_{ij} \cdot \langle \nabla f \rangle_i \right) \hat{\mathbf{B}}_i : \left(\mathbf{e}_{ij} \otimes \nabla W_{ij} \right), \tag{11}$$

where $\mathbf{e}_{ij} = \mathbf{r}_{ij}/r_{ij}$ and $\hat{\mathbf{B}}_i$ is a truncation tensor obtained by solving the linear system $\hat{\mathbf{B}}_i : \hat{\mathbf{C}}_i = -\mathbf{I}$ with $\hat{\mathbf{C}}_i$ being the fourth-order tensor:

$$\hat{\mathbf{C}}_i = \sum_{j \in I} V_j \mathbf{r}_{ij} \otimes \mathbf{e}_{ij} \otimes \mathbf{e}_{ij} \otimes \nabla W_{ij} + \left(\sum_{j \in I} V_j \mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \otimes \nabla W_{ij} \right) \cdot \mathbf{B}_i \cdot \left(\sum_{j \in I} V_j \mathbf{e}_{ij} \otimes \mathbf{e}_{ij} \otimes \nabla W_{ij} \right)$$

Due to the symmetry of $\mathbf{B}_i$ as well as the fact that for each index j, the vectors used to construct $\hat{\mathbf{C}}_i$ are elements of span($\mathbf{e}_{ij}$), one can deduce that the linear equation $\hat{\mathbf{B}}_i : \hat{\mathbf{C}}_i = -\mathbf{I}$ can be recast into a linear system of the from $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ with $\mathbf{A}$ being a symmetric matrix and $\mathbf{x}$ being the upper triangular entries of $\hat{\mathbf{B}}_i$. This implies that the dimensionality of the solution space of the linear system is reduced from d^2 to a $(d^2 + d)/2$ where d is the number of spatial dimensions of the PDE. Of course, with the Hessian tensor itself being symmetric, this result should be expected.

While it is possible to solve these linear systems via an analytical inverse in 2D, 3D simulations require solving a 6×6 linear system. For this reason, it was chosen to solve all linear systems directly following an $\mathbf{LDL}^T$ decomposition. The implication on computational cost is discussed in detail in the results section.

2.3. GFD formulation

Rather than basing the first-order gradient on the classical SPH gradient, Lanson and Vila's work [34, 35] introduced a mesh-free finite-difference based gradient discretisation. This was further expanded upon by Basic et al. [36], where a mesh-free finite-difference based Laplacian discretisation was introduced.

By using a Taylor series expansion of a generic analytic function f and applying a scaling strategy based on the distance to a nearby interpolation point:

$$W_{ij} \mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \cdot \nabla f_i = W_{ij} \mathbf{r}_{ij} \left((f_i - f_j) + \mathcal{O}(r_{ij}^2) \right), \tag{12}$$

it becomes possible to recover the renormalised mesh-free gradient via a summation of (12) for each particle with a label in I. The obtained gradient approximation can be introduced into a second-order Taylor series expansion. Following this, a first-order accurate Laplacian discretisation is obtained. This discretisation has the benefit of only requiring the same truncation tensor as the gradient discretisation, and so only requires one linear system to be solved per evaluation point.

The GFD gradient and Laplacian approximations are given as:

$$\langle \nabla f \rangle_i = \mathbf{B}_i \cdot \sum_{j \in I} (f_i - f_j) W_{ij} \mathbf{r}_{ij}, \tag{13}$$

$$\left\langle \nabla^2 f \right\rangle_i = 2d \frac{\sum_{j \in I} (f_j - f_i) W_{ij} \left(1 - \mathbf{r}_{ij} \cdot \mathbf{B}_i \cdot \mathbf{o}_i\right)}{\sum_{j \in I} r_{ij}^2 W_{ij} \left(1 - \mathbf{r}_{ij} \cdot \mathbf{B}_i \cdot \mathbf{o}_i\right)}, \tag{14}$$

where the truncation tensor $\mathbf{B}_i$ and offset vector $\mathbf{o}_i$ is given as:

$$\mathbf{B}_i = \left(\sum_{j \in I} W_{ij} \mathbf{r}_{ij} \otimes \mathbf{r}_{ij} \right)^{-1}, \tag{15}$$

$$\mathbf{o}_i = \sum_{j \in I} W_{ij} \mathbf{r}_{ij}. \tag{16}$$

To keep this scheme consistent with the SPH scheme, matrix inversion is again performed directly by making use of $\mathbf{LDL}^T$ decomposition.

To allow the comparisons between schemes to highlight the differences in response due to the differential operator discretisations only, the kernel function W_{ij} along with the scaling distance h are identical to those used in the SPH scheme for all cases considered.

Regarding the implementation details for both the gradient and Laplacian operators, it is important to note the following optimisations introduced. When considering (13), the linear contraction $\mathbf{B}_i \cdot \mathbf{r}_{ij}$ can be performed after summation. This essentially mimics the procedure followed by SPH gradient truncation. In a similar spirit, instead of saving $\mathbf{o}_i$, the contraction $\mathbf{B}_i \cdot \mathbf{o}_i$ can be precomputed and stored before being used in (14).

2.4. Mass correction

The WC formulation has been used extensively in SPH for mass-momentum coupling as it is a computationally inexpensive scheme with a simple implementation, especially when compared to projection methods used in truly incompressible formulations. It has, however, been shown that vanilla implementations of WCSPH can lead to spurious oscillations in the fluid pressure field [64].

To correct this behaviour, a modification to the discretised continuity equation as presented in [65] is used to dampen the oscillations. The modified mass equation is given as:

$$\frac{d\rho_i}{dt} = -\rho_i \left(\langle \nabla \cdot \mathbf{u} \rangle_i - \Gamma \mathcal{D}_i \Delta t \right), \tag{17}$$

with:

$$\mathcal{D}_i = \left\langle \nabla \cdot \left(\frac{\nabla p}{\rho} \right) \right\rangle_i - \left\langle \nabla \cdot \left\langle \frac{\nabla p}{\rho} \right\rangle \right\rangle_i \tag{18}$$

where Γ is a damping constant larger than 1. Due to the additional term being a difference between two Laplacian discretisations, and so leading to this term being of the order of the discretisation error, as well as being scaled by the time-step size Δt, it has been shown that (17) rapidly tends towards the unmodified continuity equation.

The SPH approximation of divergence of the pressure gradient is given by:

$$\left\langle \nabla \cdot \left(\frac{\nabla p}{\rho} \right) \right\rangle_i = 2 \sum_{j \in I, i \neq j} V_j \left(\frac{2(p_i - p_j)}{(\rho_i + \rho_j) r_{ij}} - \mathbf{e}_{ij} \cdot \langle \nabla p \rangle_i \right) \hat{\mathbf{B}}_i : \left(\mathbf{e}_{ij} \otimes \nabla W_{ij} \right), \tag{19}$$

while the SPH approximation of divergence of the SPH approximation of the pressure gradient is given by:

$$\left\langle \nabla \cdot \left\langle \frac{\nabla p}{\rho} \right\rangle \right\rangle_i = \sum_{j \in I} V_j \left(\left\langle \frac{\nabla p}{\rho} \right\rangle_j - \left\langle \frac{\nabla p}{\rho} \right\rangle_i \right) \cdot \mathbf{B}_i \cdot \nabla W_{ij} \tag{20}$$

For the case of the GFD formulation, the divergence of the pressure gradient is given by:

$$\left\langle \nabla \cdot \left(\frac{\nabla p}{\rho} \right) \right\rangle_i = 2d \frac{\sum_{j \in I} (p_j/\rho_j - p_i/\rho_i) W_{ij} \left(1 - \mathbf{r}_{ij} \cdot \mathbf{B}_i \cdot \mathbf{o}_i\right)}{\sum_{j \in I} r_{ij}^2 W_{ij} \left(1 - \mathbf{r}_{ij} \cdot \mathbf{B}_i \cdot \mathbf{o}_i\right)} \tag{21}$$

Similarly to the SPH case, the GFD approximation of divergence of the GFD approximation of the pressure gradient is given by:

$$\left\langle \nabla \cdot \left\langle \frac{\nabla p}{\rho} \right\rangle \right\rangle_i = \sum_{j \in I} W_{ij} \left(\left\langle \frac{\nabla p}{\rho} \right\rangle_i - \left\langle \frac{\nabla p}{\rho} \right\rangle_j \right) \cdot \mathbf{B}_i \cdot \mathbf{r}_{ij} \tag{22}$$

2.5. Particle shifting

It has been found that numerical instabilities may arise if Lagrangian particles evolve in an unconstrained manner based on the appropriate discretisation of the NSE alone. These numerical instabilities are caused by particles' tendency to form clusters while advecting along converging streamlines [17, 20, 65].

A solution to this problem was proposed in [20], whereby the degree of clustering around a particle is used as a measure to determine to what extent the particles should be redistributed.

Each particle is shifted according to the following scheme:

$$\delta \mathbf{r}_i = \beta \|\mathbf{u}\|_{max} \Delta t \sum_{j \in I} \frac{\bar{r}_i^2}{r_{ij}^2} \mathbf{e}_{ij}, \tag{23}$$

where β is a decrowding factor set between 0.001 and 0.1, $\|\mathbf{u}\|_{max}$ is the maximum fluid velocity, Δt is the time step size, $\bar{r}_i$ is the average distance to all particles with labels in I and $\delta \mathbf{r}_i$ being the displacement of the shift for the particle.

To keep the fields at least first-order accurate, the particle values are updated with the following scheme:

$$\delta f_i = \langle \nabla f \rangle_i \cdot \delta \mathbf{r}_i, \tag{24}$$

where f is either the pressure or velocity fields. The density is modified with a similar approach. However, (3) is now used to determine the update based on the pressure variation:

$$\delta \rho_i = \delta p_i \frac{d\rho_i}{dp_i} = \left(\frac{\rho_0}{\rho_i} \right)^{\gamma - 1} \frac{\delta p_i}{c_s^2} \tag{25}$$

2.6. Integration scheme

A two-step predictor-corrector integrator scheme is used to perform the time integration of the system. The scheme is given as:

$$f_i^* = f_i^n + \Delta t \frac{df_i^n}{dt} \tag{26}$$

$$f_i^{**} = f_i^n + \Delta t \frac{df_i^*}{dt} \tag{27}$$

$$f_i^{n+1} = \frac{1}{2}(f_i^{**} + f_i^*) \tag{28}$$

where f is either density or velocity, f^* is the predictor variable, f^{**} is the corrector variable and the time derivatives are obtained from the appropriate discretisation of (1) and (2) based on the specific MLM given as:

$$\frac{d\rho_i^n}{dt} = -\rho_i^n \left(\langle \nabla \cdot \mathbf{u}^n \rangle_i - \Gamma \mathcal{D}_i^n \Delta t \right) \tag{29}$$

$$\frac{d\mathbf{u}_i^n}{dt} = -\left\langle \frac{\nabla p^n}{\rho^n} \right\rangle_i + \nu \left\langle \nabla^2 \mathbf{u}^n \right\rangle_i + \mathbf{g}_i \tag{30}$$

Of course, $\frac{df_i^*}{dt}$ is a re-evaluation of (29) and (30) obtained by using the predictor values $\mathbf{u}^*$, ρ^* and p^* being the pressure obtained from ρ^*.

Positions are only updated at the end of the iteration using the new fluid velocities:

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t \mathbf{u}_i^{n+1} \tag{31}$$

This implies that it is only required to solve the linear systems for either of the considered MLMs at the beginning of each time step.

As mentioned in Section 2.1, the speed of sound enforces a limit on the time-step size. This is combined with the rate of momentum diffusion to generate the following Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t = s \min \left(\frac{\delta_{min}}{c_s + \|\mathbf{u}\|_{max}}, \frac{\delta_{min}^2}{\nu} \right), \tag{32}$$

where δ_{min} is the smallest particle spacing, $\|\mathbf{u}\|_{max}$ is the maximum fluid velocity, ν is the kinematic viscosity and s is a user-specified stability factor. For the simulation considered in this study, s was kept below 0.25 since this was found to be stable over all simulations considered and being comparable to similar stability conditions used in other studies [66–69].

2.7. Boundary conditions

In this section, the treatment of both solid and free surface boundaries are discussed. Due to the triviality of fixed velocity conditions at solid boundaries, more focus is dedicated to pressure boundary conditions and free surface boundary conditions.

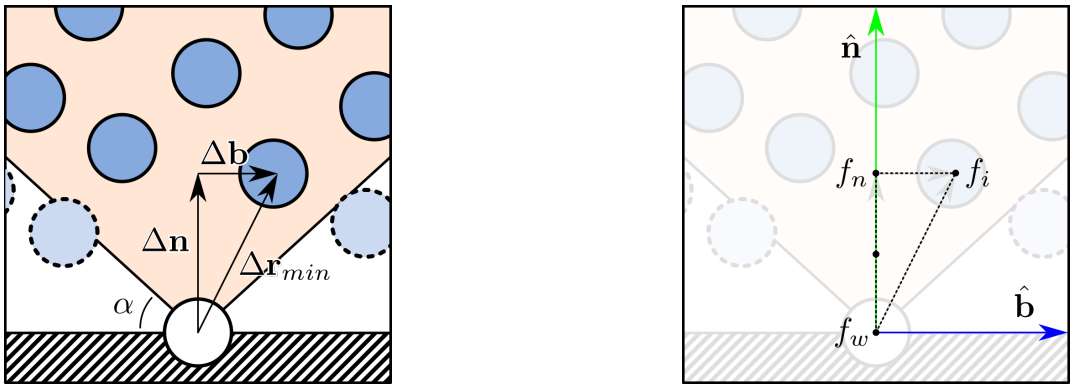
Solid boundary Neumann conditions are enforced with a hybrid MLM-FD scheme where, by setting the boundary function values, the appropriate surface gradients are obtained.

For the free surface boundaries, a scheme is proposed that allows for the appropriate generation of free surface particles. These free surface particles are assigned pressure and velocity values to enforce the required boundary conditions.

2.7.1. Solid boundaries

To enforce solid surface pressure boundary conditions, care is taken to consistently assign pressure values so that the NSE is still satisfied on boundary nodes. Due to the reduced accuracy of MLM differential operator approximations at boundaries, an extension to the hybrid MLM-FD approximation described in [37] is used to determine surface gradient information.

Figure 1 presents a schematic reference for the boundary condition implementation. Here, α is an angle that defines a cone that extends from the boundary particle used to identify valid fluid particles. Any fluid particles found outside of this cone are considered invalid. The cone angle is a heuristic parameter of the scheme. The trade-off between having valid particles spatially close to the surface normal and the robustness of the particle detection must be considered when choosing α. In this work, it was found that $\alpha = 30°$ provided the desired behaviour. Once valid particles are identified, the nearest fluid particle is determined. The distance to the nearest fluid particle is denoted by $\Delta \mathbf{r}_{min}$. This information is then used to extrapolate the field information to the boundary. For an arbitrary analytic function f, by using a Taylor series expansion around the nearest



(a) (b)

Figure 1: Schematic description of boundary condition implementation showing (a) decomposition of $\Delta\mathbf{r}_{min}$ into its normal $\Delta\mathbf{n}$ and normal complement $\Delta\mathbf{b}$ components and (b) the point labels.

fluid neighbour point, the value of f directly normal to the boundary node f_n is obtained from:

$$f_n = f_i - \Delta\mathbf{b} \cdot \langle \nabla f \rangle_i \tag{33}$$

Here, the subscript i indicates the value at the fluid neighbour while $\Delta\mathbf{b} = \Delta\mathbf{r}_{min} - \Delta\mathbf{n}$ is the complement of the projection of $\Delta\mathbf{r}_{min}$ onto the surface normal $\hat{\mathbf{n}}$ as seen in Figure 1a. Clearly then, this defines $\Delta\mathbf{n} = (\Delta\mathbf{r}_{min} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} = (\Delta n)\hat{\mathbf{n}}$. This also describes the normal point $\mathbf{r}_n$ for any boundary point $\mathbf{r}_w$ as $\mathbf{r}_n = \mathbf{r}_w + \Delta\mathbf{n}$.

It should be noted that this translation is kept only first-order accurate implying $\langle \nabla f \rangle_i = \langle \nabla f \rangle_n$.

When considering a central finite difference scheme, clearly this gives an approximation to the gradient at the mid-point between $\mathbf{r}_i$ and $\mathbf{r}_w$ as shown in Figure 1b. Since only gradients in the normal direction are required for this method, it now becomes possible to couple a central finite difference gradient with the MLM gradient via extrapolation:

$$\left.\frac{\partial f}{\partial n}\right|_w = 2\frac{f_n - f_w}{\Delta n} - \hat{\mathbf{n}} \cdot \langle \nabla f \rangle_n \,, \tag{34}$$

where the subscript w indicates a variable at the wall. With this machinery in place, it now becomes possible to enforce Neumann boundary conditions by setting the value of f_w as:

$$f_w = f_n - \frac{\Delta n}{2}\left(\hat{\mathbf{n}} \cdot \langle \nabla f \rangle_n + \left.\frac{\partial f}{\partial n}\right|_w\right) \tag{35}$$

This can now be applied to pressure boundary conditions on solid surfaces. By projecting (2) onto the surface normal, the following pressure boundary condition is obtained:

$$\frac{1}{\rho_w}\left.\frac{\partial p}{\partial n}\right|_w = \nu \left\langle \nabla^2 \mathbf{u} \right\rangle_w \cdot \hat{\mathbf{n}} + \mathbf{g} \cdot \hat{\mathbf{n}} - \frac{d\mathbf{u}_w}{dt} \cdot \hat{\mathbf{n}} \tag{36}$$

For generality, the boundary acceleration term is included. This is obtained from the solid surface's acceleration directly. However, with all boundaries considered in this study remaining stationary, $d\mathbf{u}_w/dt = 0$. Although not influencing this study's results, the surface acceleration becomes important when dealing with dynamic fluid-solid interactions. When considering the viscous contributions, since $\hat{\mathbf{n}}$ is treated as a constant vector for each specific boundary particle, this implies that $\left\langle \nabla^2 \mathbf{u} \right\rangle_w \cdot \hat{\mathbf{n}}$ can also be written as $\left\langle \nabla^2 \eta \right\rangle_w$ where η is the flow in the normal direction. It should be emphasised that this formulation is agnostic of the global surface structure. Smooth and non-smooth surfaces may be treated with this method provided that the changes in the surface normal between boundary particles are captured adequately with a fine enough discretisation. Although

only planar faces have been used in this work, this has been shown to be the case by treating the face, edge and corner features with the same scheme.

As mentioned in [17], the term $\nu \left\langle \nabla^2 \eta \right\rangle_w$ has a negligible influence in the case of non-impinging flows. However, this term becomes important to prevent boundary penetration for the case of violent fluid-boundary interactions. An implication of the previous statement makes it possible to identify that this term's dominating component is focussed in the normal direction. As such, to only deal with the critical terms, the Laplacian is recast into its local surface coordinate components as a method to extract $\partial_{nn}\eta$ specifically. Following a Taylor series expansion of η in the normal direction, combined with (34), this term can be written as:

$$\left.\frac{\partial^2 \eta}{\partial n^2}\right|_w = \frac{2}{\Delta n^2}\left(\eta_n - \eta_w - \Delta n \left.\frac{\partial \eta}{\partial n}\right|_w\right) \tag{37}$$

By combining (35) and (37) for the case of pressure boundary conditions specifically, (36) can now be written as:

$$p_w = p_n - \frac{\rho_w \Delta n}{2}\left(\hat{\mathbf{n}} \cdot \left\langle \frac{\nabla p}{\rho} \right\rangle_n + \nu \left.\frac{\partial^2 \eta}{\partial n^2}\right|_w + \mathbf{g} \cdot \hat{\mathbf{n}} - \frac{d\mathbf{u}_w}{dt} \cdot \hat{\mathbf{n}}\right) \tag{38}$$

It should be noted that ideally p_w and ρ_w are coupled. However, due to the small variation in ρ_w, for this study it was chosen to use the previous iteration's value of ρ_w to compute the new pressure.

With abrupt changes in highly non-linear fields, the sudden jump between nearest neighbours may cause issues to the numerical stability of the scheme. Smoothing of this response can be achieved by carrying out the above procedure for several valid particles in the support radius and building a weighted approximation:

$$p_w = \frac{\sum_{i \in I} p_{w,i} W_{iw}}{\sum_{i \in I} W_{iw}}, \tag{39}$$

where $p_{w,i}$ is the pressure obtained by (38) for the i^{th} particle. It should be noted that this was not required for the simulations considered in this study and so we use the limiting case of only considering the single nearest particle.

This process is carried out for all boundary particles on the surface of the solid interface. When resolving this interface, the boundary particles are generated to ensure the spacing is consistent with the fluid configuration with initial spacing δ_0. With the geometries of the cases investigated in this work being constructed from simple line segments and rectangular faces in 2D and 3D cases, respectively, Cartesian meshes with spacing δ_0 are used to generate boundary particle positions.

It should also be noted that this method not only imposes a Neumann pressure boundary condition consistent with the NSE, but also does so by determining an appropriate surface pressure value. This allows the scheme to be extended for applications in resolved fluid-solid coupling schemes where the forcing conditions on the solid particles are partially determined by surface pressure. For a detailed example on these coupling strategies in the context of SPH simulations, we direct the reader to the work of Hashemi et al. [17] for a 2D implementation or Joubert et al. [25] for a 3D implementation.

2.7.2. Free surface boundaries

In the standard SPH formulation, free surfaces are handled naturally with the construction of the kernel approximations at a reduced support essentially enforcing the free surface conditions.

This, however, ceases to be the case for first-order accurate approximations. Due to the truncation process extrapolating internal information into areas with reduced support, free surface conditions are no longer naturally enforced. For a WC system, a solution proposed by Oger et al. [16] uses a hybrid approach. The truncation tensor is used to determine free surface particles after which an alternative SPH approximation is applied to these particles to automatically enforce the boundary conditions. In ISPH, where a linear system is solved to determine fluid particle pressures, the free surface conditions are explicitly enforced by linear equations of the discretised pressure Poisson equation [22]. With some modifications, due to the use of kernel correction, it may be possible to apply similar techniques to a WC system as well. This, however, requires one to identify the free surface particles as well.

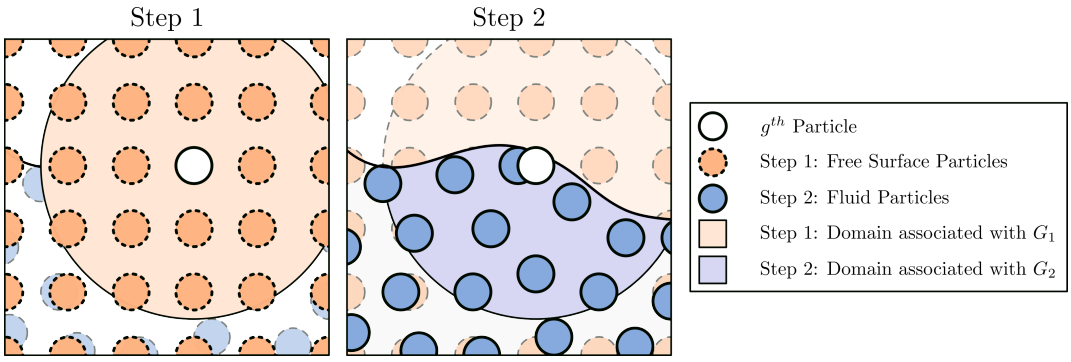


Figure 2: Domains of interest for steps 1 and 2 of the free surface boundary enforcement.

A novel method to apply a free surface boundary condition is proposed with the intent of mimicking the information that standard SPH considers by using an underlying fixed mesh coupled with a volume measure as a way to obtain a layer of free surface particles that surround and support particles at free surfaces. This has the added benefit of always ensuring free surface particles have neighbours, making the building of truncation tensors more robust.

To place a free surface particle, a two-step process is used. Both steps use a volume measure given as:

$$\mu_g^k = \sum_{i \in G_k} W_{ig}, \tag{40}$$

where g indicates a specific free surface particle and G_k indexes a set of neighbouring particles, where k depends on which step is considered.

As shown in Figure 2, for step 1, G_1 indexes the set of all neighbouring particles in the grid while for step 2, G_2 indexes the set of all neighbouring fluid particles. Of course, since the mesh is regular, step 1 can be reduced to only determining μ_g^1 near boundaries since internal particles will have a constant μ_g^1 that can be determined during initialisation. For the case where both the geometry remains fixed and there is no memory constraint, the boundary values of μ_g^1 can also be precomputed during initialisation as well.

For particles sufficiently far from the fluid domain, $\mu_g^2 = 0$. For this reason, a search over the bounding domain is performed to identify zones that contain fluid particles and so potentially free surface particles as well. μ_g^2 is only determined in these zones.

Finally, a particle is flagged as a free surface particle if $0 < \mu_g^2 < s\mu_g^1$ where s is a scaling constant. For the cases presented, it was found that $s = 0.35$ provides the desired response.

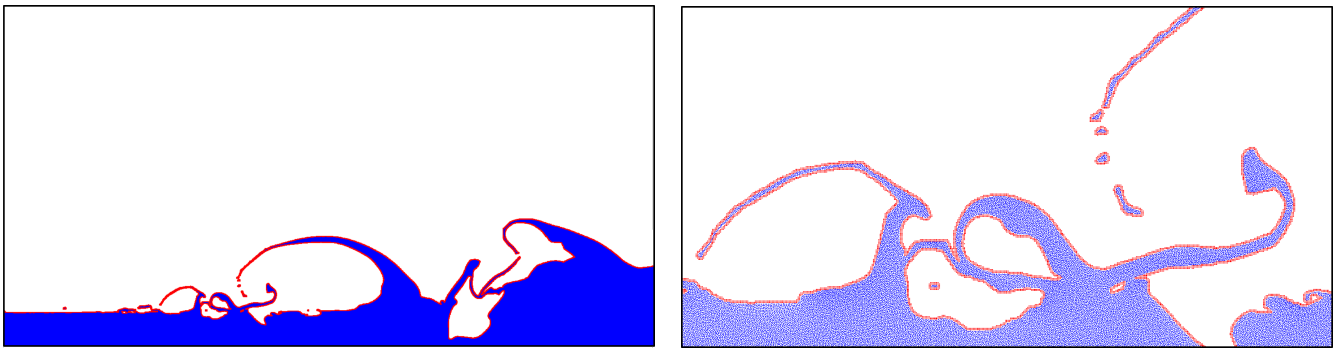
It should be noted that since these particles lie on a regular mesh, their positions do not need to be saved explicitly, reducing the memory requirements of this method drastically.

An example of this method's free surface system is presented in Figure 3, where the blue and red particles indicate the fluid and active free surface particles, respectively.

Pressure boundary conditions are enforced by setting the pressure of the free surface particles to 0. To mimic the free surface system moving with the fluid particles, the velocity of free surface particles should be the same as the free surface velocity. The nearest fluid particle to each free surface particle is determined during the evaluation of μ_g^2. The velocities of the free surface particles are then linked to their fluid pairs.

All kernel approximations remain as presented in the above sections by treating the free surface particles as other fluid particles.

Finally, it should be noted that this method also has an application in the explicit enforcement of pressure boundary conditions



(a) (b)

Figure 3: Free surface shown in red obtained from the corresponding MLM system shown in blue for (a) the full system, and (b) a zoomed-in section.

as seen in [22] due to the required identification of the nearest fluid particle to these free surface particles, which identifies a set of boundary particles of the fluid particle set.

2.8. Implementation details

The flow diagram of both MLM schemes for the 3D solver is presented in Figure 4. The process in blue represents the host code that runs on the CPU, while red processes represent device code that runs on the GPU. The CPU handles only the initialisation of the solver. This includes input file reading, initial fluid and free surface particle generation and meshing of the boundary. For device processes, parallelisation is handled at the particle level. This includes neighbour search, boundary conditions enforcement, differential operator evaluation and integration.

The neighbour search makes use of the spatial hash map approach used very commonly in SPH and MLMs in general to identify nearby particles. A cell size of the kernel support radius is used to generate a grid over the computational domain. A particle is assigned a hash value unique to the cell it lies within. Neighbouring particles are found by performing a search over all cells neighbouring the current particle cell.

With the Laplacian's dependency on the truncation tensors and gradients, the scheme requires a second pass over the neighbouring particles. For the 3D solver, the truncation tensor is grouped with the evaluation of the pre-truncated gradients. This allows the distance and kernel function evaluations for the gradients to be reused in the truncation tensor construction. The pre-truncated gradients for the SPH solver are the zeroth-order SPH gradients. For the GFD scheme, this is the summation of the weighted finite difference terms before the linear operator is applied. To obtain first-order accurate gradients, the truncation tensors are inverted and applied to the gradient terms.

With the predictor-corrector integration scheme, the differential operators are evaluated twice. Since the particle positions are not updated during the predictor step, the truncation tensors are not re-computed. As described in Section 2.6, once the next time-step's density and velocity values are determined, the particle positions are updated based on the new velocity. Boundary condition enforcement set the next time-step's solid boundary particle information by reusing the differential operators obtained during the integration step.

Except for the truncation tensors' construction, the 2D and 3D solvers follow an identical implementation. In the 2D solver's case, the truncation tensor is built independently from the gradients thereby requiring a third pass over all neighbouring particles. This was done to allow the computational cost associated with the tensor construction to be measured in isolation. With the increased number of distance and kernel function evaluations in 3D, this code optimisation has a drastically enhanced effect on the computational cost of 3D simulations and so could not justifiably be ignored.

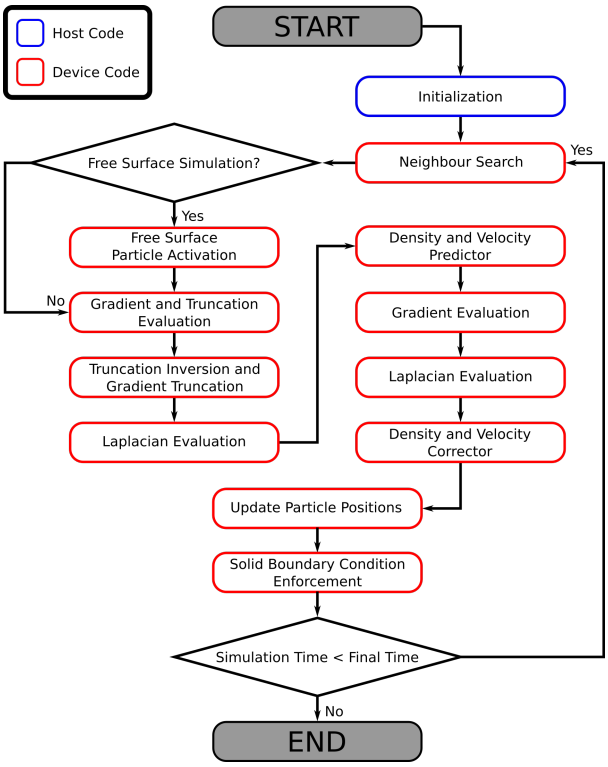


Figure 4: Flow diagram of the numerical scheme.

3. Results

In the following sections, WCSPH and WCGFD schemes are compared over three different problem sets that have traditionally served as test cases for MLMs and CFD in general. Specifically, the lid-driven cavity and the dam-break systems are considered. The lid-driven cavity is only examined in a 2D setting while the dam-break is investigated in both a 2D and 3D configuration. These cases have been chosen as they highlight the application of the proposed boundary conditions methods to no-slip, free surface and symmetry boundary conditions. The computational performance of both MLMs is also investigated. The obtained results are also assessed critically to highlight any key differences between each method's solutions.

The solvers were implemented in C++ using the NVIDIA CUDA 10.1 toolkit for parallelisation. The NVIDIA GeForce 1080 GTX graphics card was used to obtain the results and performance metrics in the sections below. The 2D solver was implemented in its own independent framework, while the 3D solver was implemented inside the GPU based DEM framework Blaze-DEM [70–72].

3.1. 2D Lid-driven cavity

The first comparison between the behaviour of the two solvers is considered in the context of fully submerged confined flow. For this, the classical lid-driven cavity is considered. Here, a square cavity filled with a viscous fluid subject to the boundary conditions presented in Figure 5 is simulated.

Validation of the solvers is performed at both $Re = 100$ and $Re = 1000$. The lid velocity is set to $U_{lid} = 1\text{m/s}$, while the kinematic viscosity is chosen to be $\nu = 0.1\text{m}^2/\text{s}$. The cavity lengths are then constrained to $H = 10\text{m}$ and $H = 100\text{m}$. For the $Re = 100$ case, an initial spacing of $\delta_0 = 0.025\text{m}$ is used while the $Re = 1000$ case uses $\delta_0 = 0.25\text{m}$. This leads to both fluids being discretised by 160k particles.

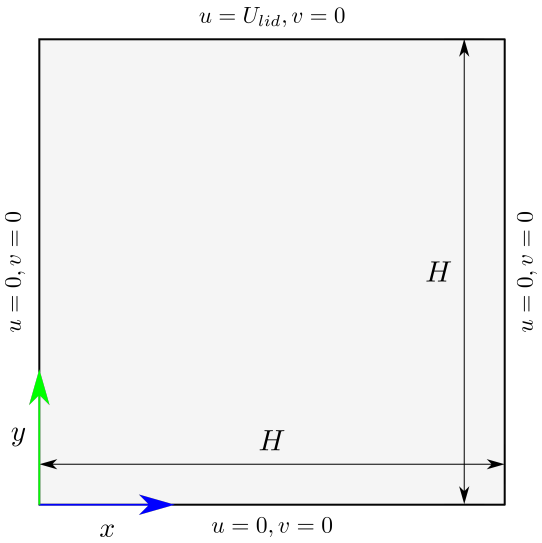


Figure 5: Schematic representation of the lid-driven cavity problem.

	$Re = 100$ Ghia et al.	$Re = 100$ Xu et al.	$Re = 1000$ Ghia et al.	$Re = 1000$ Xu et al.
x-error (SPH) [%]	0.249	0.653	0.207	1.985
x-error (GFD) [%]	0.244	0.761	0.262	2.124
y-error (SPH) [%]	0.744	1.888	0.532	3.215
y-error (GFD) [%]	0.759	2.061	0.681	3.459

Table 1: Mean percentage difference between mid-plane velocities of obtained results and the work of Ghia et al. [73] and Xu et al. [20]

The mid-plane velocities obtained from WCSPH and WCGFD can be seen in Figures 6a and 6b. The classical results of Ghia et al. [73] and the ISPH results of Xu et al. [20] at a resolution of 161×161 particles are presented as comparisons. Table 1 shows the mean normalised difference between the current results and those from literature. Normalisation is performed relative to the maximum range of the mid-plane velocity considered.

As discussed in [74], the naive no-slip SPH boundary conditions produce lower energy solutions corresponding to the observed lower velocities at the higher Reynolds numbers of the ISPH results. This issue does not arise in either the WCSPH or WCGFD simulations by applying the MLM-FD hybrid boundary conditions.

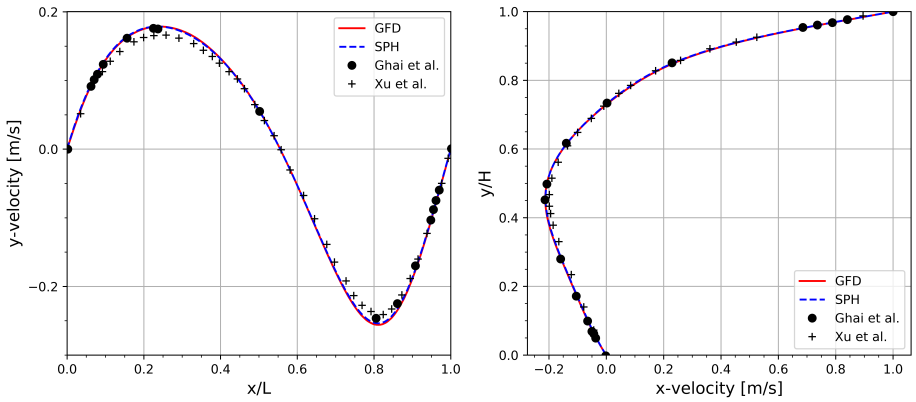
For a more in-depth comparison between the WCGFD and WCSPH solvers, both solvers' response to a varying system resolution is investigated. Again $U_{lid} = 1\text{m/s}$, $\nu = 0.1\text{m}^2/\text{s}$ and $H = 10\text{m}$ leading to $Re = 100$. The initial spacing and the total number of particles used in each simulation can be seen in Table 2.

The steady-state velocity magnitude for the 4 different cases for the WCGFD simulations and the differences between the WCSPH and WCGFD solutions can be seen in Figures 7a to 7d. The mid-plane velocities can be seen in Figure 8.

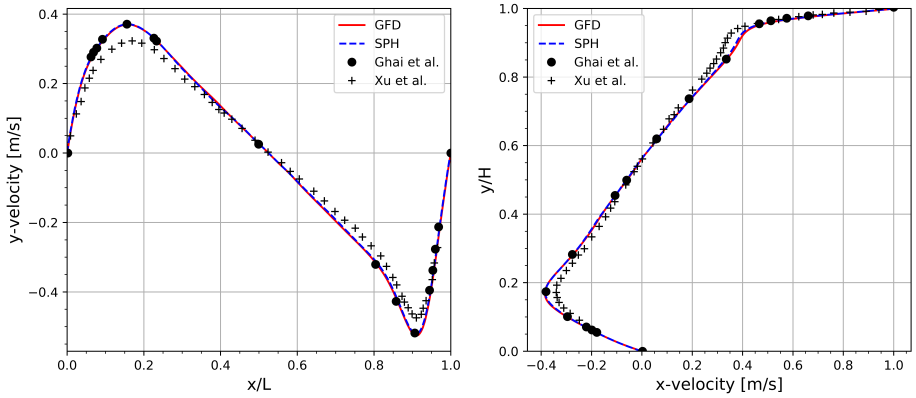
For the field response, it can be seen that as the resolution increases, the overall error decreases. It can also be noticed that larger errors become more isolated to specific regions in the domain. When comparing mid-plane velocities, it can be seen that both solvers predict very similar responses even for coarser simulations where the solution is still mesh-dependent. Figure

δ_0 [m]	0.2	0.1	0.05	0.025
Particle number	2.5k	10k	40k	160k

Table 2: Particle count for Section 3.1.



(a)



(b)

Figure 6: Mid-plane velocities for the lid-driven cavity at (a) $Re = 100$ and (b) $Re = 1000$ compared against the results of Ghia et al. [73] and Xu et al. [20]

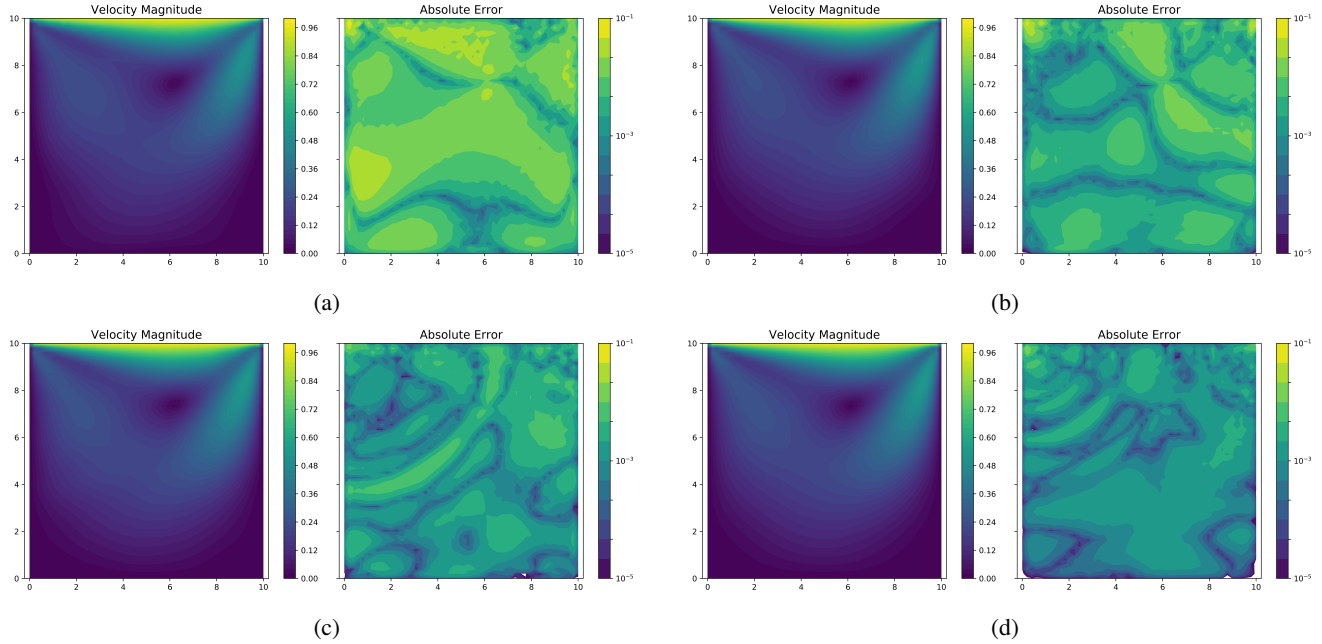


Figure 7: Contours of velocity magnitude of the WCGFD solver and absolute differences between the WCGFD and WCSPH solutions for the lid-driven cavity with (a) $\delta_0 = 0.2$m, (b) $\delta_0 = 0.1$m, (c) $\delta_0 = 0.05$m and (d) $\delta_0 = 0.025$m.

9 shows the mean normalised difference in mid-plane velocities for the two solvers over the four cases. From this, it is clear that both difference functions are monotonically decreasing. For the $\delta_0 = 0.2$m case, a mean difference of 0.98% and 1.49% for the x- and y-velocity was found, respectively, while for the $\delta_0 = 0.025$m case, a mean difference of 0.13% and 0.26% for the x- and y-velocity was found, respectively.

Figure 10a presents the computational time per iteration for both methods for the cases investigated, while Figure 10b shows the relative speedup obtained by using of the GFD formulation. A maximum speedup of 2.6x was obtained when comparing WCGFD to WCSPH.

The slope in log-log space characterises the exponential growth of the computational cost per particle count. Considering only the last section between the $\delta_0 = 0.05$ and $\delta_0 = 0.025$ cases of Figure 10a, a slope of 1.50 and 1.42 was found for WCSPH and WCGFD, respectively. While this does indicate that WCGFD's computational cost grows slower than WCSPH, considering how close these values are, it should be assumed that the cost of each scheme scales in a similar manner until a more detailed study is done on the topic.

An overview of the resource distribution during the simulations can be seen in Figure 11. The computational cost per iteration of the GPU kernels of interest for both the WCGFD and WCSPH solvers is normalised based on the time per iteration of the WCGFD method at each Reynolds number considered.

Additionally, Table 3 shows the resource distribution while running the $\delta_0 = 0.025$m simulations for both methods. As expected, both results attribute much of the performance gains to the much more efficient construction of a truncation operator, leading to most of the resources being used to resolve the differential operators. Furthermore, when considering the total time spent constructing the differential operators, GFD is more than 2x as efficient in these operations. Relative to the SPH truncation inversions, one can see nearly a factor 10 speedup to the solving of the linear systems; however, due to the small size of these linear systems, this is not a critical factor in the computational cost or performance gains.

When considering the coarser cases, it can also be noticed that the boundary enforcement contributes up to 13.2% of the computational cost. As the resolution becomes finer, this becomes dwarfed by the sheer number of fluid particles relative to boundary particles. This does nevertheless show that the boundary condition enforcement can incur a significant cost in cases

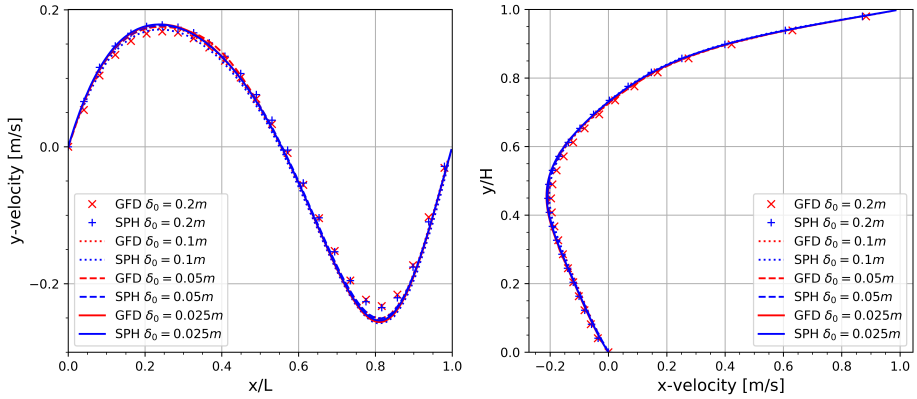


Figure 8: Mid-plane velocities for the lid-driven cavity for $\delta_0 = 0.2$m to $\delta_0 = 0.025$m.

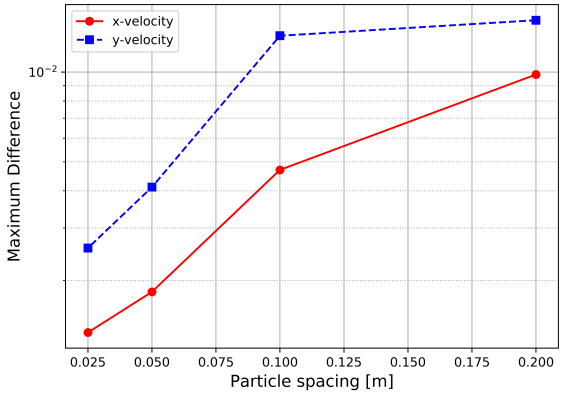
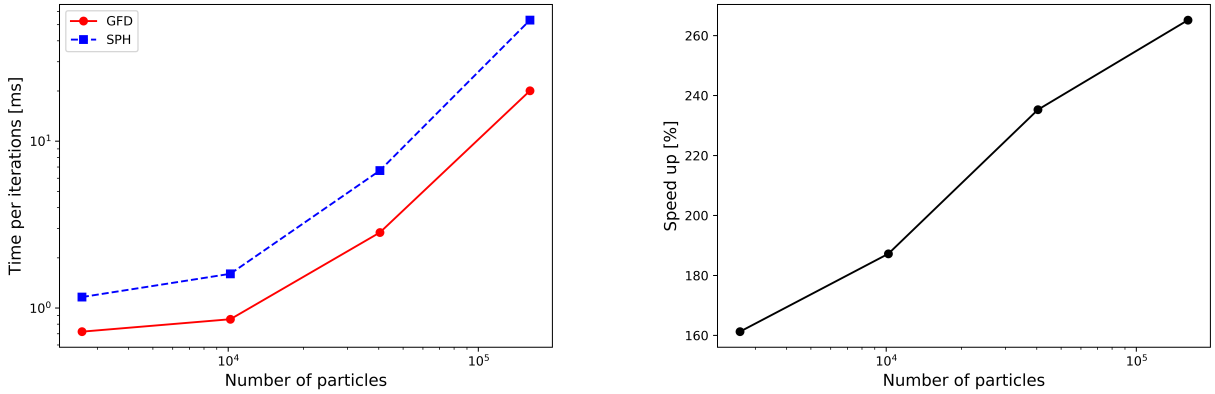


Figure 9: Mean normalised difference between WCGFD and WCSPH mid-plane velocity results.



(a) (b)

Figure 10: WCSPH and WCGFD performance comparison for the lid-driven cavity cases showing (a) the time per iteration of WCSPH and WCGFD and (b) percentage speedup of WCGFD compared to WCSPH.

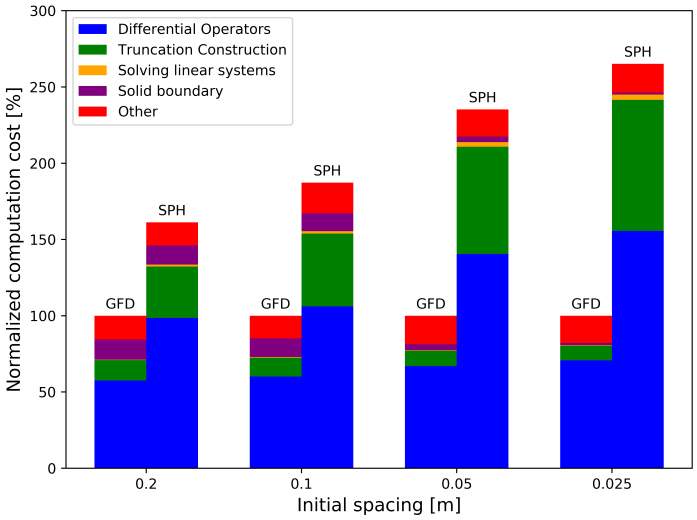


Figure 11: Normalised resource distribution of the WCGFD and WCSPH solvers.

	Time per iteration [ms]	Differential operators [ms]	Truncation construction [ms]	Solving linear system [ms]
SPH	53.305	31.253	17.266	0.677
GFD	20.101	14.216	1.930	0.070

Table 3: Resource distribution for $\delta_0 = 0.025$ lid-driven cavity case.

with complex geometries where the surface area of the system is large compared to its volume.

A similar computational cost is expected for either scheme's gradient discretisations. Specifically, both require the construction of the gradient truncation tensor $\mathbf{B}^{-1}$ followed by a linear mapping of the weighted sum of difference terms. The key difference between these schemes is the weighting term used. For SPH, this is given by $V_j \nabla W_{ij}$ while for GFD this is given by $W_{ij}\mathbf{r}_{ij}$. The computational demands to obtain either are similar.

This implies that the speedup is mainly due to the Laplacian discretisation. The first major improvement comes from completely avoiding the construction of the fourth-order tensor $\hat{\mathbf{C}}_i$ in the GFD scheme. The construction of the Laplacian itself is also less demanding. Due to the memory constraints imposed by the GPU, the summation of the Laplacian is obtained by resolving the full scalar value for each particle pair. This implies that the SPH scheme must resolve the dot product $\mathbf{e}_{ij} \cdot \langle \nabla f \rangle_i$ and the linear contractions $\hat{\mathbf{B}}_i : \left(\mathbf{e}_{ij} \otimes \nabla W_{ij}\right)$. With $\mathbf{B}_i \cdot \mathbf{o}_i$ being pre-computed, GFD only requires the leading dot product of $\mathbf{r}_{ij} \cdot \mathbf{B}_i \cdot \mathbf{o}_i$ to be resolved per particle pair.

At a low level, this implies that the SPH Laplacian requires more floating-point operations (FLOPs) per particle pair. To quantify this, an approximation of the construction process for the differential operators as presented in [75] is considered. The approximation treats the k^{th} update step of the full system state as a matrix-vector multiplication operation $\mathbf{F}^{k+1} = \mathbf{G}^k \cdot \mathbf{F}^k$ where $\mathbf{G}^k$ is a sparse $N \times N$ matrix and $\mathbf{F}^k$ is the state vector of the particle system. However, unlike the solver proposed in [75], the performance benefits of GFD does not come from reducing the number of nearest neighbours but rather by reducing the number of operations per particle pair and so the cost of constructing $\mathbf{G}^k$ must be evaluated rather than the cost of operation $\mathbf{G}^k \cdot \mathbf{F}^k$. For this reason, this approximation is applied to each differential operator and truncation tensor independently. To include the more expensive operations, the cost of linear operators is resolved before applying this approximation. This implies that linear contractions are treated as an additional addition and multiplication operation per dimension. Any tensor object is treated as d_T systems where d_T is the number of independent entries of the tensor.

For a single field, to resolve the matrix multiplication requires $2nN$ operations where N is the number of particles and n is the bandwidth of the matrix given by the average number of neighbouring particles. The number of scalar fields N_{DOF} give the degrees of freedom for a particle. For both SPH and GFD, the gradients require $(2nN)dN_{DOF}$ operations while either gradient truncation tensor requires $(2nN)d_s$ operations where d is the dimensionality of the system and $d_s = (d^2 + d)/2$ is the number of unique entries for the symmetric 2-tensor.

The SPH Laplacian requires $2d_s(nN)$ operations to resolve the tensor contraction and $2dN_{DOF}(nN)$ operations for the dot products. A single dot product must be evaluated per particle pair for the GFD Laplacian. With the numerator and denominator both being constructed simultaneously, this requires $N_{DOF} + 1$ matrix-vector multiplication systems.

The SPH Laplacian truncations construction is the most expensive of the operators considered. This is due to the three sub tensors that must be pre-constructed before determining $\hat{\mathbf{C}}$. Due to symmetry the third and fourth order tensors require $d_3 = 4$ and $d_4 = 5$ unique entries in 2D, respectively. This increases to $d_3 = 10$ and $d_4 = 15$ in 3D. A total of $(2nN)\left(2d_3 + d_4\right)$ operations are required to construct $\hat{\mathbf{C}}$. The GFD equivalent is the construction of the offset vector requiring only $(2nN)d$ operations.

Table 4 summarises these results along with the ratio between the number of SPH and GFD operations in both 2D and 3D. Since the same number of particles on average are considered for both solvers, the results are normalised by $2nN$. Although this analysis provides insight into the distribution of the schemes' computational demands, the overall speedup for the truncation and differential operator evaluations by this estimate is only 2.0 in 2D. This is well below the obtained ratio of 3.0. A source of error comes from approximating all intermediate scalar FLOPs at a particle pair level by a single multiplication operator essentially requiring the scaling of the $2nN$ factors associated with matrix-vector multiplication for each object. However, the estimate provided already captures the significant differences between the schemes. So while the overall FLOP complexity of both schemes will increase, they will do so in a similar manner. A more reasonable explanation for the disparity between the practical and the theoretical speedup arises when emphasising that relating the FLOP complexity to computational time assumes that the scheme makes uses of serial operations that are not memory bound. The GPU implementation leads to an additional layer of complexity with memory reading, writing and allocation playing a significant role in the performance and likely being the reason for the disparity between the practical and the theoretical speedup.

Operation	SPH	GFD	SPH/GFD (2D)	SPH/GFD (2D)
Gradient	dN_{DOF}	dN_{DOF}	1.0	1.0
Laplacian	$d_s + (1+d)N_{DOF}$	$d + 1 + N_{DOF}$	2.0	2.75
Gradient truncation	d_s	d_s	1.0	1.0
Laplacian truncation	$2d_3 + d_4$	d	6.5	11.67

Table 4: Cost per particle pair for differential operator construction.

3.2. 2D Dam-break

This section considers the response of a classic dam-break case frequently used to test SPH solvers. A schematic representation of this problem can be seen in Figure 12 with $H = h = 0.6$m. In this study, the same dam-break case is investigated with varying particle resolutions. Table 5 shows the initial spacing and total particle count for each of the cases considered.

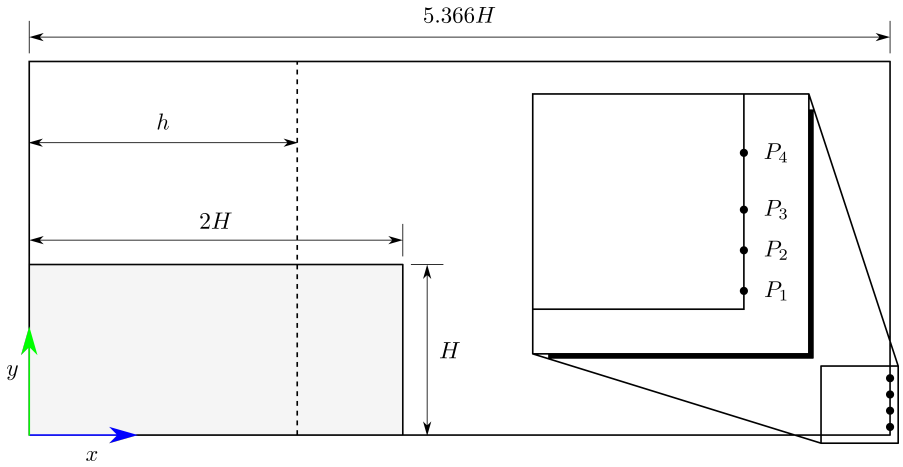


Figure 12: Schematic representation of the dam-break problem.

First, the solvers are validated by comparing the results obtained to the experimental work of Lobovský et al. [76] at the maximum resolution considered. A gravitational acceleration of $g = -9.81\text{m/s}^2$ is used along with a kinematic viscosity of $\nu = 5 \times 10^{-4}\text{m}^2/\text{s}$. Comparisons between the wavefront position, fluid column height at $x = h$ and wall pressure at P_1 to P_4 are used to validate the response. The pressure sensors are located on the end wall at heights of $0.01H$, $0.05H$, $0.1H$ and $0.267H$. Figure 14 shows the boundary particle positions and two lower particle sensor locations. The pressure sensor values are determined from the two neighbouring boundary particles via linear interpolation of their pressure values.

Figure 13 shows snapshots of the velocity magnitude of the fluid for the WCGFD solver. Figure 15 shows the dam-break's kinematic response over time while Figure 16 shows the surface wall pressure over time at the pressure sensors considered.

When considering the kinematic results, good agreement can be seen between the experimental and numerical results. Both the WCSPH and WCGFD wavefront lies within a 0.12 range of the experimental results with an average error of the order 0.05. The non-dimensional steady wavefront velocity of 1.62 for the experimental results compares well to the numerical predictions of 1.61 as well. The predicted column height also compares well with experimental results with the results lying within a 0.035 range with an average error of the order 0.02.

With the highly dynamic nature of the flow after impact, a one-to-one match as with the kinematic results is not to be expected for the pressure field. However, it can be seen that the overall trends in the pressure response also closely match experiments.

A similar problem is considered for the comparison between the solvers. The system geometry is kept identical to that used during the validation case. A key difference here is that a kinematic viscosity of $\nu = 5 \times 10^{-3}\text{m}^2/\text{s}$ is now used. With a

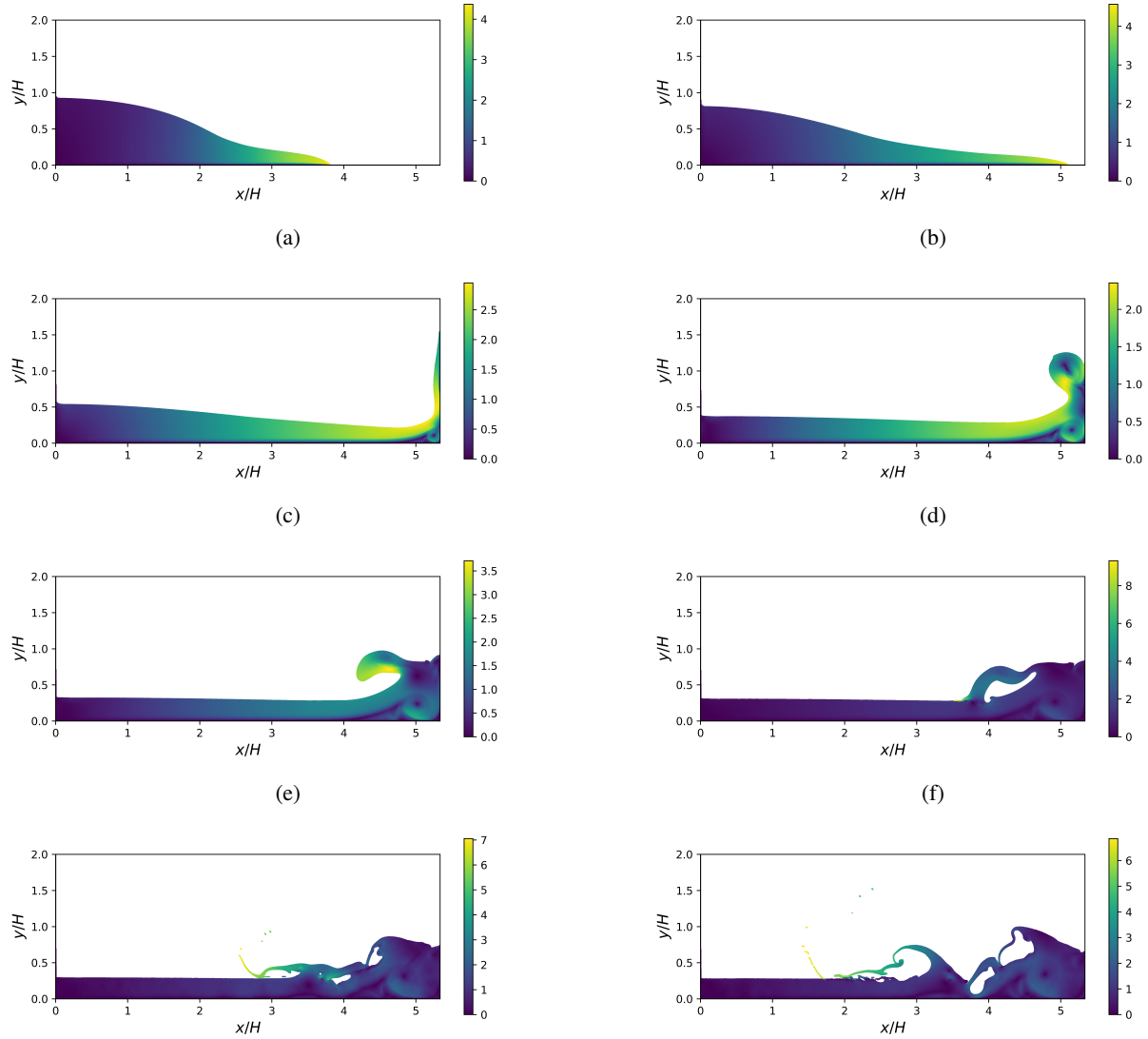


Figure 13: Contours of velocity magnitude of the WCGFD solver at (a) $t\sqrt{g/H} = 1.617$, (b) $t\sqrt{g/H} = 2.426$, (c) $t\sqrt{g/H} = 4.044$, (c) $t\sqrt{g/H} = 5.661$, (e) $t\sqrt{g/H} = 6.470$, (f) $t\sqrt{g/H} = 6.874$ (g) $t\sqrt{g/H} = 7.278$ and (h) $t\sqrt{g/H} = 7.683$.

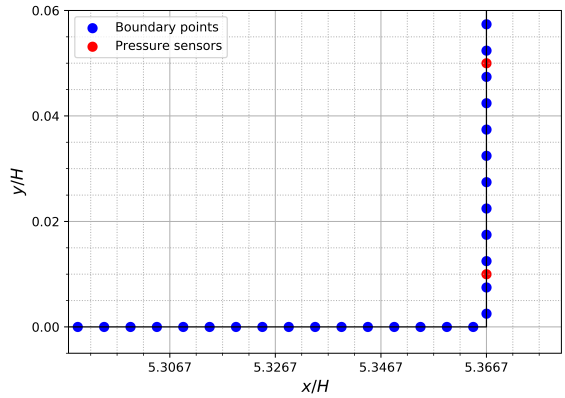
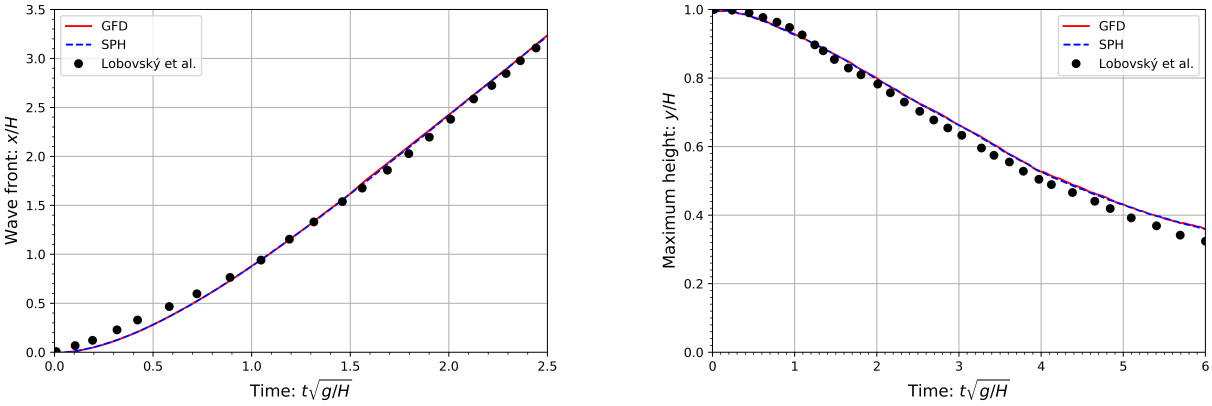


Figure 14: Boundary particle positions and pressure sensor locations.



(a) (b)

Figure 15: WCSPH and WCGFD results for (a) wavefront and (b) column height compared against the results of Lobovský et al. [76].

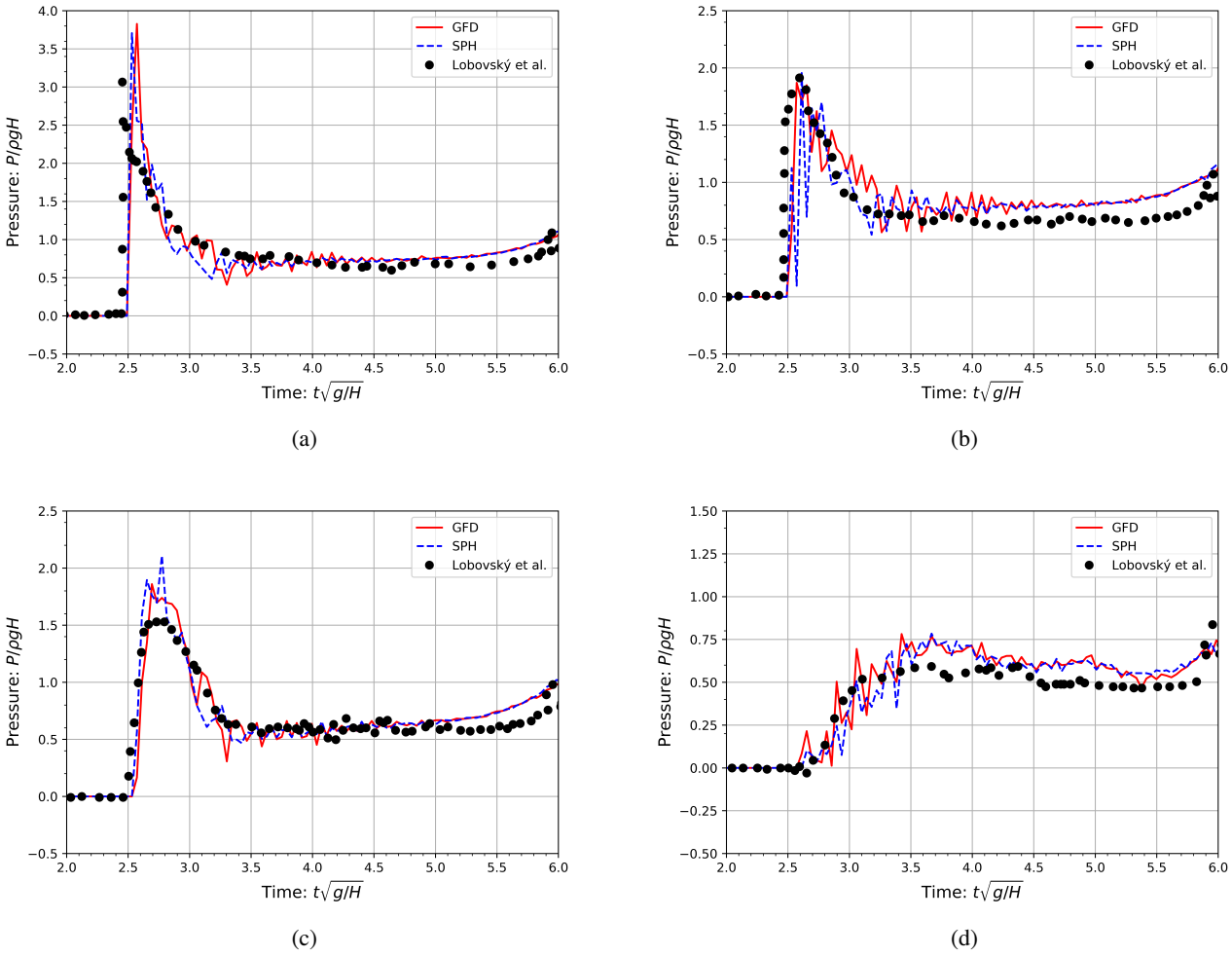
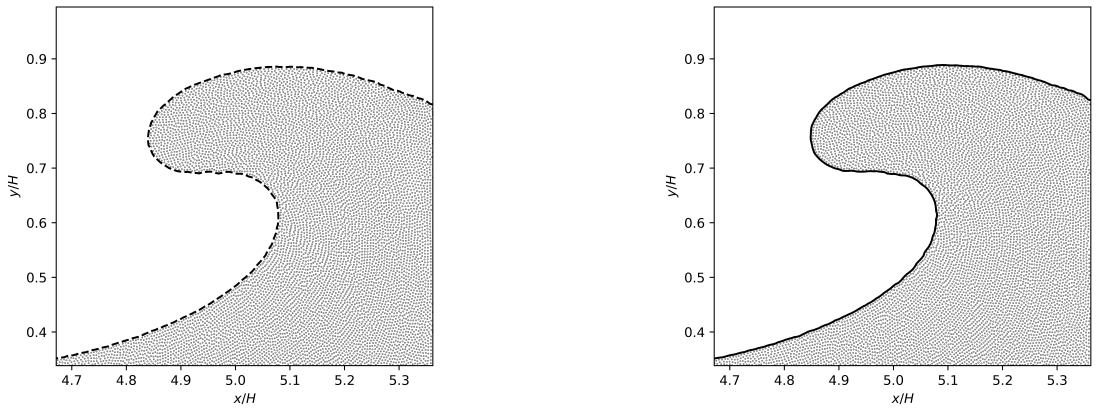


Figure 16: Wall pressure for WCSPH and WCGFD at (a) P_1, (b) P_2, (c) P_3 and (d) P_4 compared against the results of Lobovský et al. [76].



(a) (b)

Figure 17: Particles positions and corresponding free surface for (a) WCSPH and (b) WCGFD at $t\sqrt{g/H} = 5.85$.

δ_0 [mm]	6	4.5	3
Particle number	21.5k	37.4k	82.2k

Table 5: Particle count for Section 3.2.

more pronounced effect of momentum diffusion, this generates results with a high sensitivity to both the pressure gradient discretisation and the velocity Laplacian discretisation. It also prevents the system from evolving into a highly chaotic state after impacting the end wall allowing for a more direct description of the differences between the responses to be obtained.

For the finest case, the free surface boundary is determined at $t\sqrt{g/H} = 5.85$ and $t\sqrt{g/H} = 7.60$. For the $t\sqrt{g/H} = 5.85$ case, the particles and corresponding free surfaces can be seen in Figure 17. Similarly, Figure 18 presents the same results for the $t\sqrt{g/H} = 7.60$ case. The comparisons between these profiles can be seen in Figures 19 and 20 for $t\sqrt{g/H} = 5.85$ and $t\sqrt{g/H} = 7.60$ cases, respectively. Two measures are used to compare the results, specifically a velocity magnitude and velocity direction difference, given as:

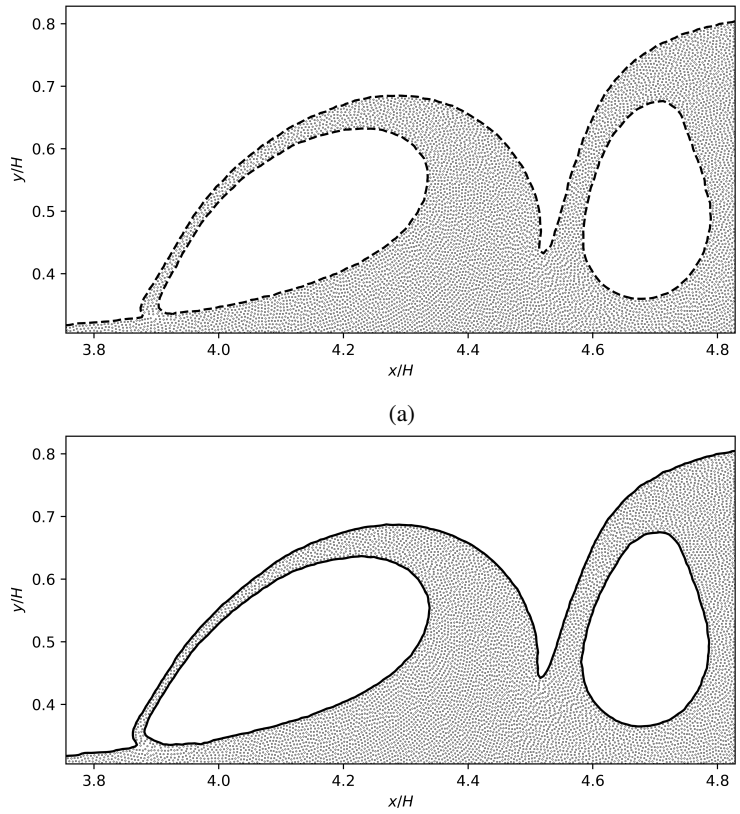
$$e_u = \left| \|\mathbf{u}_{GFD}\|_2 - \|\mathbf{u}_{SPH}\|_2 \right|, \tag{41}$$

$$e_\theta = arccos\left(\frac{\mathbf{u}_{GFD} \cdot \mathbf{u}_{SPH}}{\|\mathbf{u}_{GFD}\|_2 \|\mathbf{u}_{SPH}\|_2} \right), \tag{42}$$

where $\mathbf{u}_{SPH}$ and $\mathbf{u}_{GFD}$ are the WCSPH and WCGFD velocity fields, respectively.

While slight variations can be seen at $t\sqrt{g/H} = 5.85$, the comparison show very good agreement between the two methodologies. When looking at the $t\sqrt{g/H} = 7.60$ case, while most of the boundary shows a good agreement between the WCGFD and WCSPH results, a much more distinct difference in the boundaries can be seen on the leftmost wave. This hints at the idea that WCSPH and WCGFD deal with sufficiently supported particles in a very similar way. However, under reduced support, WCGFD is shown to be more energetic than WCSPH. In fully supported regions, it can be seen that the velocity magnitudes internally are very similar. In contrast, the velocity directions are more varied showing that internal flow differences are generated more from deviations in the fluid trajectories rather than differences between the kinetic energies of the fluids.

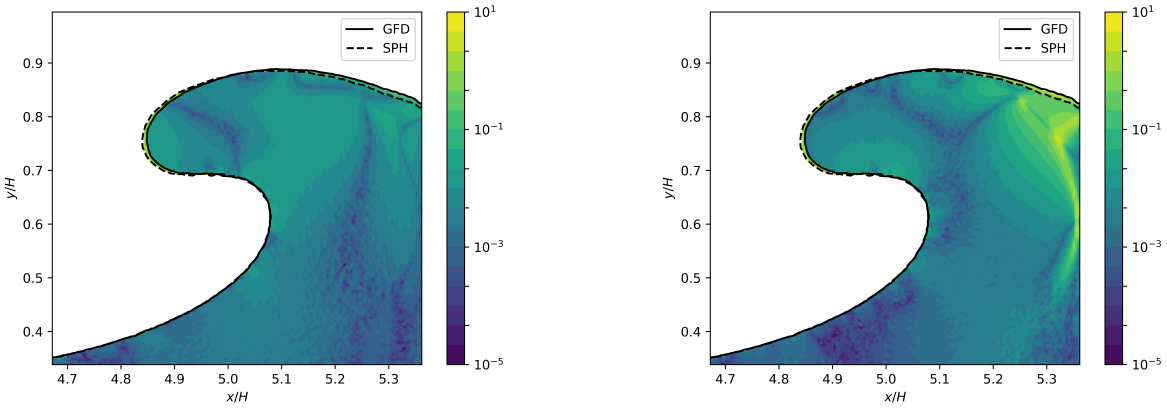
The distance travelled by the wavefront for the three cases can be seen in Figure 21a, while the height of the fluid at the position $x = h$ can be seen in Figure 21b.



(a)

(b)

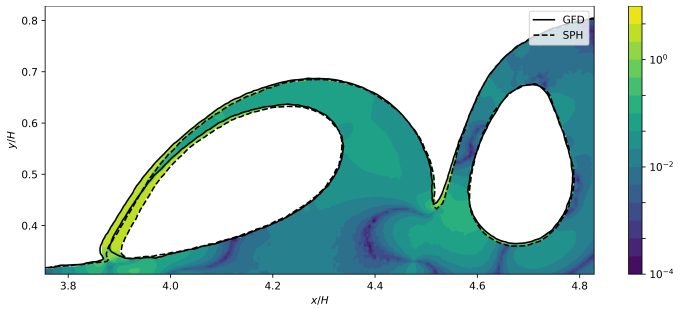
Figure 18: Particles positions and corresponding free surface for (a) WCSPH and (b) WCGFD at $t\sqrt{g/H} = 7.60$.



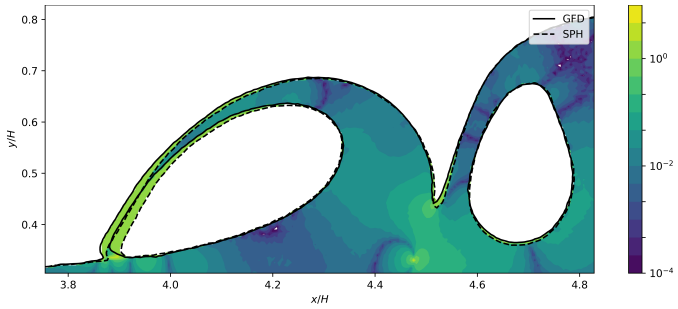
(a)

(b)

Figure 19: Comparison between WCSPH and WCGFD free surfaces at $t\sqrt{g/H} = 5.85$ in terms of (a) velocity magnitude and (b) velocity direction.

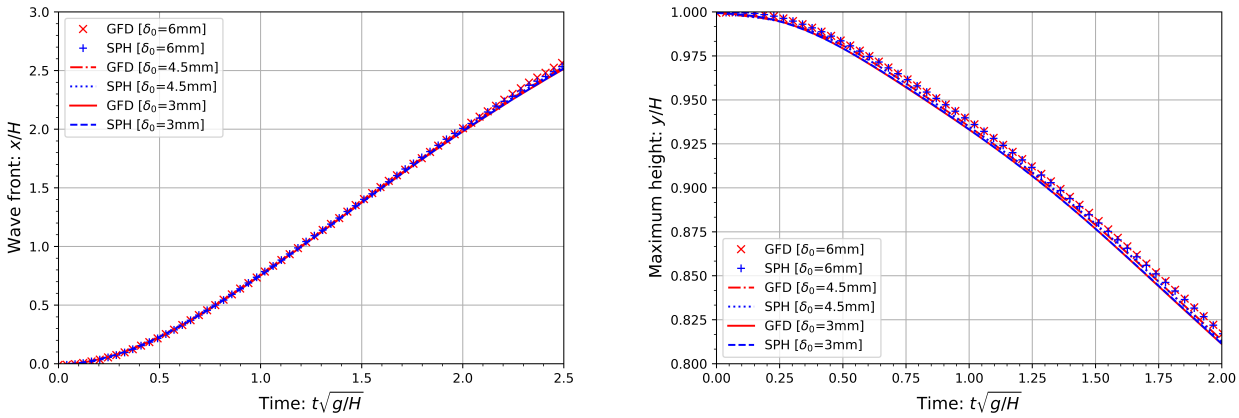


(a)



(b)

Figure 20: Comparison between WCSPH and WCGFD free surfaces at $t\sqrt{g/H} = 7.60$ in terms of (a) velocity magnitude and (b) velocity direction.



(a) (b)

Figure 21: WCSPH and WCGFD results for (a) wavefront evolution and (b) fluid height.

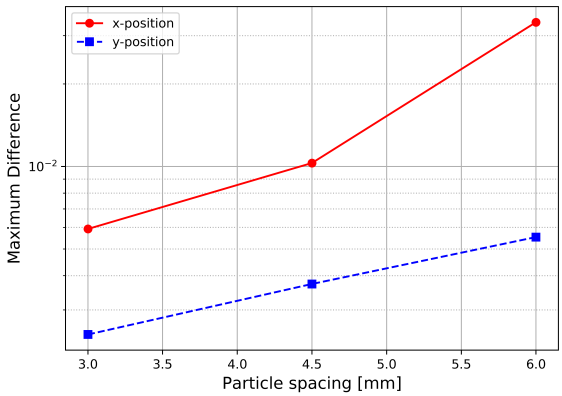
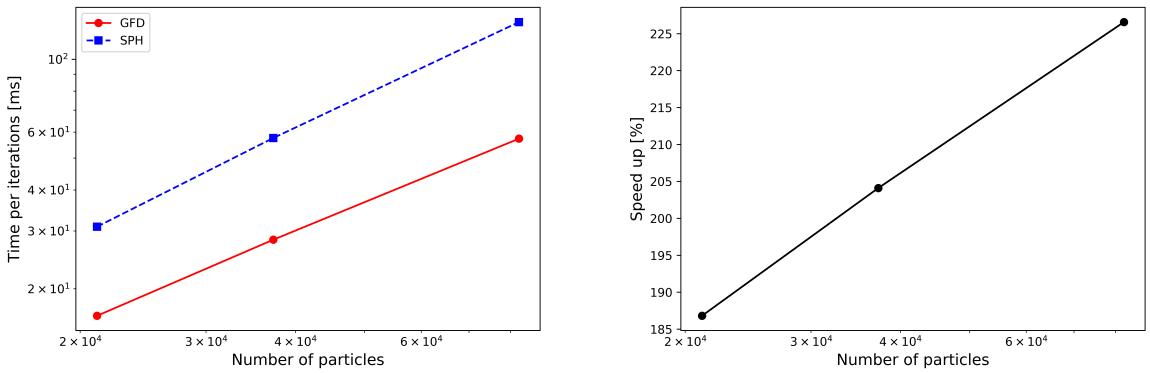


Figure 22: Maximum difference between WCSPH and WCGFD results for wavefront position x/H and fluid height y/H.



(a) (b)

Figure 23: WCSPH and WCGFD performance comparison for the dam-break cases showing (a) the time per iteration of WCSPH and WCGFD and (b) percentage speedup of WCGFD compared to WCSPH.

Again, it can be seen that there is good agreement between the WCSPH and WCGFD solvers. The difference in these results at the different fluid resolutions can be seen in Figure 22. From this, it was found that the maximum difference of 3.36×10^{-2} and 5.53×10^{-3} was found for the wavefront position and fluid height at the coarsest fluid resolution, respectively, while a maximum difference of 5.92×10^{-3} and 2.44×10^{-3} was obtained for the wavefront position and fluid height at the finest fluid resolution, respectively.

A comparison between the solver speeds is made for the three different resolutions. The results can be seen in Figure 23a, while the percentage speedup can be seen in Figure 23b. As one might expect, a similar trend as seen in section 3.1 is observed here as well. A maximum speedup is obtained in the largest simulation with a speedup factor of 2.3x. This is comparable to the speedup obtained for the lid-driven cavity for a similar particle count. In this case, one can see a constant scaling in computational cost over the tested range. As with the previous section, the slopes of the graphs in Figure 23a are calculated with the WCSPH results having a slope of 1.03 while the WCGFD results have a slightly shallower slope of 0.90.

As with the lid-driven cavity, the normalised resource distribution is shown in Figure 24, while the physical resource distribution for the finest resolution case is presented in Table 6. The noticeable difference between the lid-driven cavity and the dam-break case comes down to the slightly lower percentage cost associated with the differential operators, truncation and

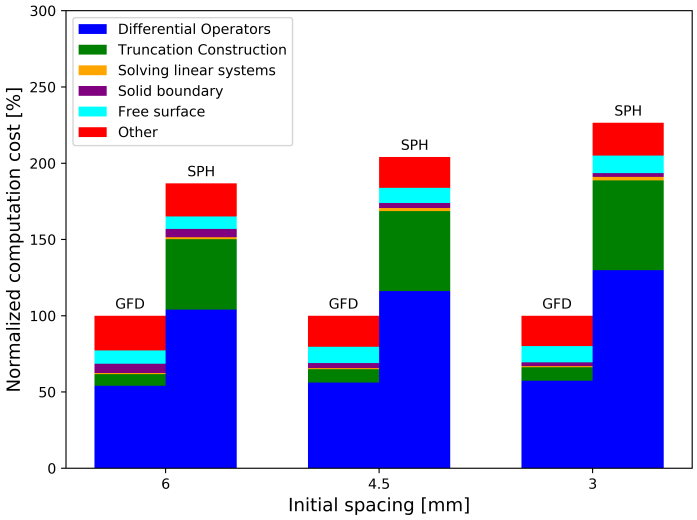


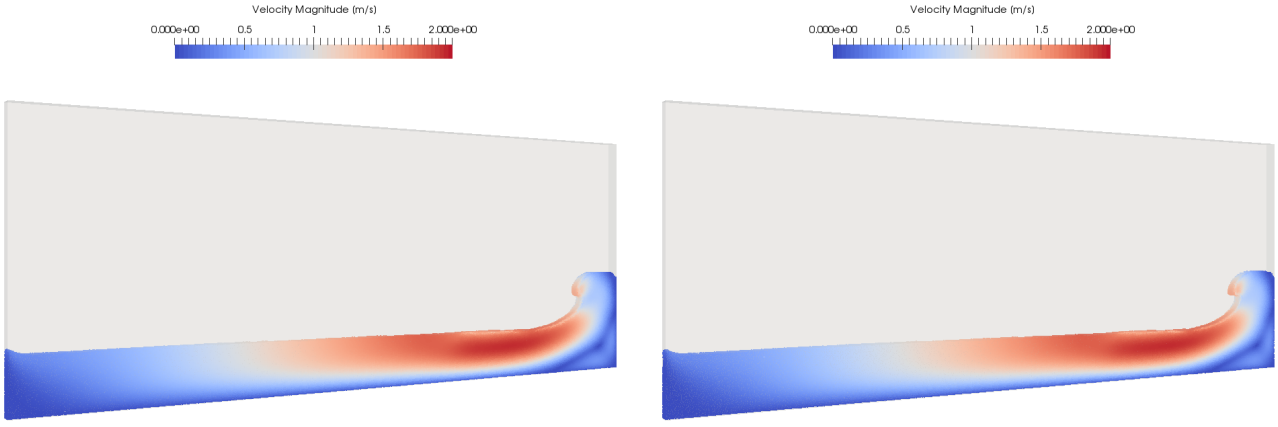
Figure 24: Normalised resource distribution of the WCGFD and WCSPH solvers for the 2D dam-break case.

	Time per iteration [ms]	Differential operators [ms]	Truncation construction [ms]	Solving linear system [ms]
SPH	25.976	14.884	6.753	0.259
GFD	11.464	6.581	1.009	0.023

Table 6: Resource distribution for the $\delta_0 = 3$mm resolution 2D dam-break case.

δ_0 [mm]	6	4.5	3
Particle number	180.0k	319.6k	720.0k

Table 7: Particle count for Section 3.3.



(a) (b)

Figure 25: Velocity magnitude of the 3D dam-break at $t\sqrt{g/H} = 5.85$ for the (a) WCSPH and (b) WCGFD methodology.

linear solver operations due to the additional cost of the free surface enforcement now also being introduced. In the current implementation, simply any free surface particle with a neighbouring fluid particle is evaluated. This is of course the reason for the near-constant percentage cost associated with the free surface kernel.

3.3. 3D Dam-break

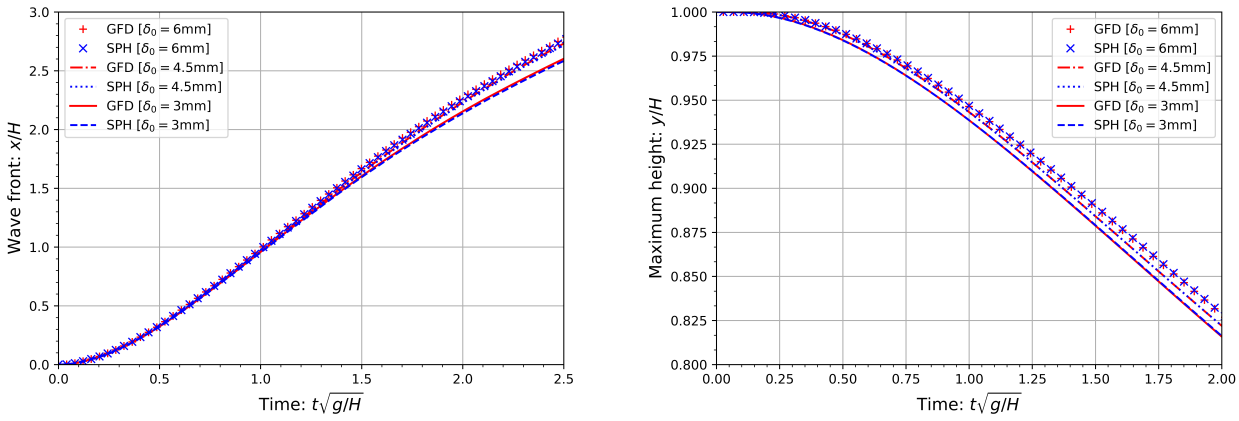
With the nonlinear dependency of computational complexity on system dimension, it is difficult to extrapolate the previous sections' findings to a higher dimensional setting. For this reason, a dam-break case similar to that of Section 3.2 is explored again in the context of a full 3D setting. Both the WCSPH and WCGFD responses are obtained from the fluid-only solvers of the computational framework Blaze-DEM.

The same system depicted in Figure 12 is constructed with the same parameters as those used in Section 3.2. The fluid properties are kept identical as well. The depth of the system is chosen in such a way as to keep the scaling of the number of particles the same as in the previous section. This implies that the depth of the geometry depends on the particle size. For this case, the depth is chosen to be $10\delta_0$. The total number of particles for each simulation can be seen in Table 7.

Apart from the two new faces, a no-slip boundary condition is enforced on all other walls. To represent the same system as before, symmetry boundary conditions are applied to the two faces parallel to the XY-plane.

Figure 25 shows a qualitative comparison between the fine case WCSPH and WCGFD simulations at $t\sqrt{g/H} = 5.85$. As expected, the responses are, from a visual point of view, nearly indistinguishable.

To quantify the differences, Figures 26a and 26b show the wavefront evolution and fluid column height, respectively. The difference between the obtained results can be seen in Figure 27. Here it can be seen that, unlike the 2D case, the 3D case still has a significant particle size dependency.



(a) (b)

Figure 26: WCSPH and WCGFD results for (a) wavefront evolution and (b) fluid height for the 3D dam-break.

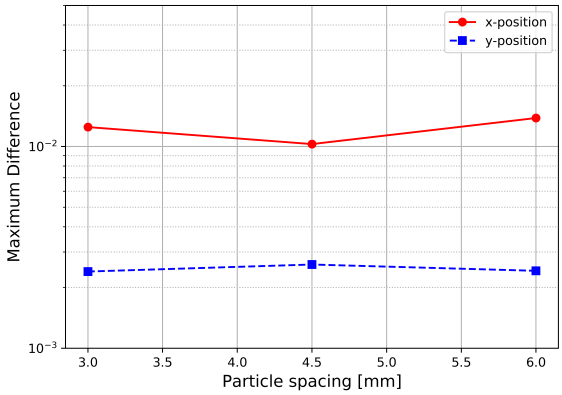


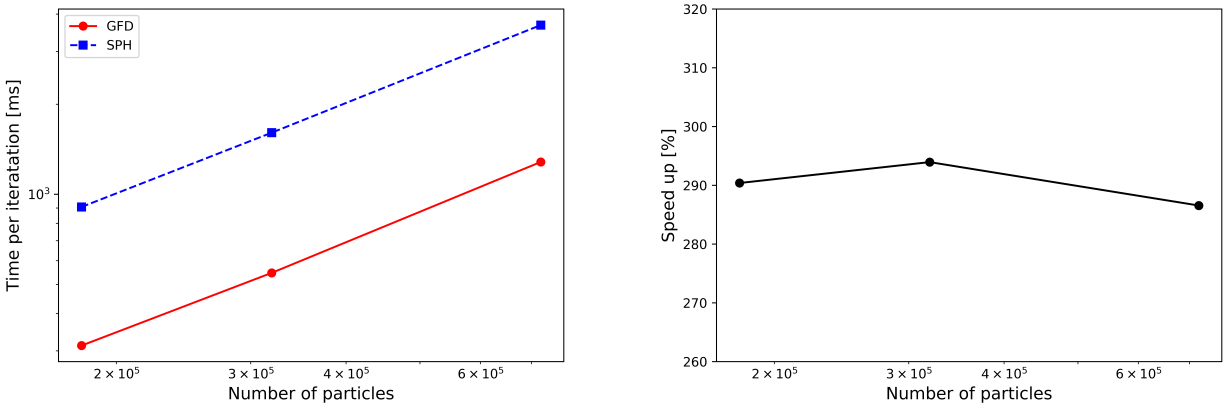
Figure 27: Maximum difference between WCSPH and WCGFD results for wave front evolution and fluid height for the 3D dam-break.

It can be noticed that the differences between the WCSPH and WCGFD simulations seems to have stagnated between a range of 8.2×10^{-3} and 1.1×10^{-2} for the wavefront and 2.4×10^{-3} and 2.6×10^{-3} for the fluid height. It is likely that errors introduced by the mesh dependency has led to this behaviour since fundamentally the key difference between these two methods are in how they approximate missing information in under-resolved areas.

To consider each method's performance, the time per iteration and speedup factor can be seen in Figures 28a and 28b, respectively. The WCGFD solver was between 2.87x to 2.94x faster than the WCSPH solver, although, unlike the 2D systems, the speedup remains fairly constant for all system resolutions considered. In this case, the plot's slope in Figure 28a for WCGFD is 1.05, which is very comparable to the WCSPH slope of 1.02.

Figure 29 shows the normalised GPU resource distribution during the simulations, while Table 8 shows the resource distribution over the course of the $\delta_0 = 3$mm simulation. Due to algorithm optimisation in Blaze-DEM, the differential operators now capture the construction of the truncation tensors as well, and so the raw cost of its construction cannot be measured directly.

When looking at the resource distributions, it can be noticed that the computational efficiency of the GFD truncation and



(a) (b)

Figure 28: WCSPH and WCGFD results for (a) wavefront evolution and (b) fluid height for the 3D dam-break.

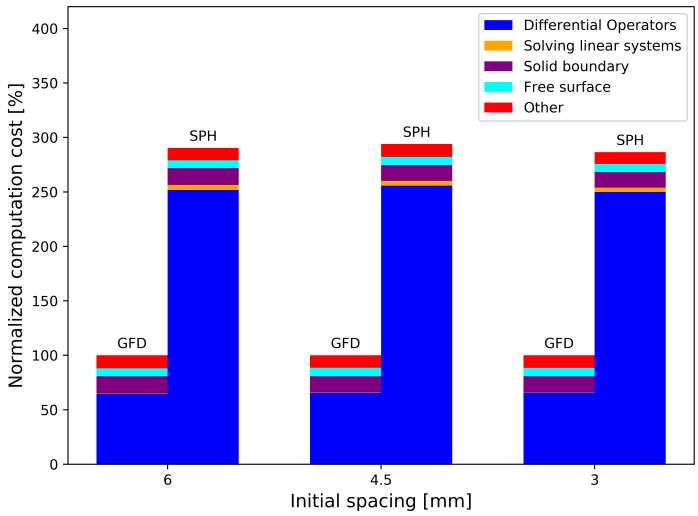


Figure 29: Normalised resource distribution of the WCGFD and WCSPH solvers for the 3D dam-break case.

	Time per iteration [ms]	Differential operators [ms]	Solving linear system [ms]
SPH	3676.41	3205.84	51.47
GFD	1282.93	842.88	5.13

Table 8: Resource distribution for $\delta_0 = 3$mm resolution 3D dam-break case.

differential operators are, as expected, a considerable factor in the speedup obtained. Due to the much smaller linear systems that are built, it can still be assumed that the construction of truncation tensors in the 3D GFD case is still drastically faster than the SPH equivalent.

A noticeable difference between the trends found in 2D and 3D is that now the normalised cost of enforcing the boundary conditions are relatively fixed. Considering that the depth of the geometry scales with the particle size and since only boundary particles on a wetted boundary do any computational work, this leads to the active boundary particle to scale consistently with the total fluid particle number.

4. Conclusion

This work demonstrated the proposed boundary condition implementations for WC renormalised MLMs in both 2D and 3D settings. Specifically for the case of solid boundary conditions, the hybrid MLM-FD technique has been shown to accurately enforce Neumann boundary conditions by considering both pressure and symmetry boundary conditions. The 2D dam-break case shows that this method accurately determines the surface pressure on the solid phase described by boundary particles. The free surface scheme has allowed WC renormalised MLMs to handle free surface flow without the need for alternative treatment of fluid particles lying on the surface. The generation of nearby surface particles has led to a more robust calculation of the renormalisation tensors, allowing smooth and fragmented free surface flow to be simulated.

In contrast to zeroth-order MLM schemes, or more specifically classical SPH, the use of first-order differential operators requires additional effort to enforce boundary conditions. It was found that, under certain conditions, a significant portion of the computational resources were dedicated to the enforcement of boundary conditions for both solvers. As shown in both the 2D and 3D dam-break cases, this cost is further increased when free surfaces are considered. It was found that this additional cost, although not as critical as the cost associated with the differential operators, still has a noticeable effect on the solver's overall performance.

When comparing the solvers' cost, it was found that the WCGFD solver was consistently more computationally efficient than the WCSPH solver. This is due to the more efficient construction of truncation operators and differential operators. This led to a maximum speedup of 2.6x and 2.9x in 2D and 3D, respectively. Furthermore, it was found that both WCGFD and WCSPH produce very similar responses for all cases considered with the results tending to converge to each other following an increase in fluid resolution. This leads to auspicious results indicating that it is possible to obtain a flat performance benefit for first-order accurate WCSPH by taking advantage of the GFD operators while not incurring a significant change in the system response.

Finally, while this paper focused on WC MLMs, there is no reason to limit the findings to WC systems in general. In terms of the differential operators, ISPH may realise similar benefits in computational efficiency when making use of GFD operators. When considering the boundary conditions, the solid boundary condition scheme is directly applicable to any renormalised MLM while the free surface method, when used as an intermediate step to identifying free surface particles, can assist in enforcing the Dirichlet pressure boundary condition of the pressure Poisson equation as well.

5. Acknowledgements

This work was supported in part by the Eiffel Excellence PhD Scholarship Program funded through the French Ministry for Europe and Foreign Affairs (Laureat 2019). We gratefully acknowledge the support of the C-AIM group, University of Pretoria for providing the NVIDIA GeForce 1080 GTX graphics cards used in this study.

References

[1] A. Colagrossi and M. Landrini. "Numerical simulation of interfacial flows by smoothed particle hydrodynamics". In: *Journal of Computational Physics* 191.2 (2003), pp. 448–475.

[2] Z. Chen, Z. Zong, M. B. Liu, L. Zou, H. T. Li, and C. Shu. "An SPH model for multiphase flows with complex interfaces and large density differences". In: *Journal of Computational Physics* 283 (2015), pp. 169–188.

[3] A. M. Tartakovsky, K. F. Ferris, and P. Meakin. "Lagrangian particle model for multiphase flows". In: *Computer Physics Communications* 180.10 (2009), pp. 1874–1881.

[4] S. Natsui, R. Nashimoto, T. Kumagai, T. Kikuchi, and R. O. Suzuki. "An SPH Study of Molten Matte-Slag Dispersion". In: *Metallurgical and Materials Transactions B: Process Metallurgy and Materials Processing Science* 48.3 (2017), pp. 1792–1806.

[5] K. Szewc, J. Pozorski, and A. Tanire. "Modeling of natural convection with Smoothed Particle Hydrodynamics: Non-Boussinesq formulation". In: *International Journal of Heat and Mass Transfer* 54.23-24 (2011), pp. 4807–4816.

[6] A. Zainali, N. Tofighi, M. S. Shadloo, and M. Yildiz. "Numerical investigation of Newtonian and non-Newtonian multiphase flows using ISPH method". In: *Computer Methods in Applied Mechanics and Engineering* 254 (2013), pp. 99–113.

[7] Z. Ji, L. Fu, X. Y. Hu, and N. A. Adams. "A new multi-resolution parallel framework for SPH". In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 1156–1178.

[8] Y. Tang, Q. Jiang, and C. Zhou. "A Lagrangian-based SPH-DEM model for fluid-solid interaction with free surface flow in two dimensions". In: *Applied Mathematical Modelling* 62 (2018), pp. 436–460.

[9] Y. He, A. E. Bayly, A. Hassanpour, F. Muller, K. Wu, and D. Yang. "A GPU-based coupled SPH-DEM method for particle-fluid flow with free surfaces". In: *Powder Technology* 338 (2018), pp. 548–562.

[10] P. W. Cleary. "Prediction of coupled particle and fluid flows using DEM and SPH". In: *Minerals Engineering* 73 (2015), pp. 85–99.

[11] W. Pan, K. Kim, M. Perego, A. M. Tartakovsky, and M. L. Parks. "Modeling electrokinetic flows by consistent implicit incompressible smoothed particle hydrodynamics". In: *Journal of Computational Physics* 334 (2017), pp. 125–144.

[12] W. Hu, G. Guo, X. Hu, D. Negrut, Z. Xu, and W. Pan. "A consistent spatially adaptive smoothed particle hydrodynamics method for fluid-structure interactions". In: *Computer Methods in Applied Mechanics and Engineering* 347 (2019), pp. 402–424.

[13] R. B. Canelas, A. J. C. Crespo, J. M. Dominguez, R. M. L. Ferreira, and M. Gómez-Gesteira. "SPH-DCDEM model for arbitrary geometries in free surface solid-fluid flows". In: *Computer Physics Communications* 202 (2016), pp. 131–140.

[14] P. W. Randles and L. D. Libersky. "Smoothed particle hydrodynamics: Some recent improvements and applications". In: *Computer Methods in Applied Mechanics and Engineering* 139.1-4 (1996), pp. 375–408.

[15] R. Fatehi and M. T. Manzari. "Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives". In: *Computers and Mathematics with Applications* 61.2 (2011), pp. 482–498.

[16] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. "An improved SPH method: Towards higher order convergence". In: *Journal of Computational Physics* 225.2 (2007), pp. 1472–1492.

[17] M. R. Hashemi, R. Fatehi, and M. T. Manzari. "A modified SPH method for simulating motion of rigid bodies in Newtonian fluid flows". In: *International Journal of Non-Linear Mechanics* 47.6 (2012), pp. 626–638.

[18] N. Trask, M. Maxey, K. Kim, M. Perego, M. L. Parks, K. Yang, and J. Xu. "A scalable consistent second-order SPH solver for unsteady low Reynolds number flows". In: *Computer Methods in Applied Mechanics and Engineering* 289 (2015), pp. 155–178.

[19] A. Khayyer, H. Gotoh, and S. D. Shao. "Corrected Incompressible SPH method for accurate water-surface tracking in breaking waves". In: *Coastal Engineering* 55.3 (2008), pp. 236–250.

[20] R. Xu, P. Stansby, and D. Laurence. "Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach". In: *Journal of Computational Physics* 228.18 (2009), pp. 6703–6725.

[21] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani. "δ-SPH model for simulating violent impact flows". In: *Computer Methods in Applied Mechanics and Engineering* 200.13-16 (2011), pp. 1526–1542.

[22] A.D. Chow, B.D. Rogers, S.J. Lind, and P.K. Stansby. “Incompressible SPH (ISPH) with fast Poisson solver on a GPU”. In: *Computer Physics Communications* 226 (2018), pp. 81–103.

[23] R. Shamsoddini and B. Abolpur. “Investigation of the Effects of Baffles on the Shallow Water Sloshing in A Rectangular Tank Using A 2D Turbulent ISPH Method”. In: *China Ocean Engineering* 33.1 (2019), pp. 94–102.

[24] W. Hu, W. Pan, M. Rakhsha, Q. Tian, H. Hu, and D. Negrut. “A consistent multi-resolution smoothed particle hydrodynamics method”. In: *Computer Methods in Applied Mechanics and Engineering* 324 (2017), pp. 278–299.

[25] J. C. Joubert, D. N. Wilke, N. Govender, P. Pizette, U. Tuzun, and N. E. Abriak. “3D gradient corrected SPH for fully resolved particle–fluid interactions”. In: *Applied Mathematical Modelling* 78 (2020), pp. 816–840.

[26] S. Koshizuka and Y. Oka. “Moving-particle semi-implicit method for fragmentation of incompressible fluid”. In: *Nuclear Science and Engineering* 123.3 (1996), pp. 421–434.

[27] A. Shakibaeinia and Y. Jin. “A weakly compressible MPS method for modeling of open-boundary free-surface flow”. In: *International Journal for Numerical Methods in Fluids* 63.10 (2010), pp. 1208–1232.

[28] A. Shakibaeinia and Y. Jin. “MPS mesh-free particle method for multiphase flows”. In: *Computer Methods in Applied Mechanics and Engineering* 229-232 (2012), pp. 13–26.

[29] T. Xu and Y. Jin. “Improvements for accuracy and stability in a weakly-compressible particle method”. In: *Computers and Fluids* 137 (2016), pp. 1–14.

[30] M. Jandaghian and A. Shakibaeinia. “An enhanced weakly-compressible MPS method for free-surface flows”. In: *Computer Methods in Applied Mechanics and Engineering* 360 (2020).

[31] A. Khayyer and H. Gotoh. “Enhancement of stability and accuracy of the moving particle semi-implicit method”. In: *Journal of Computational Physics* 230.8 (2011), pp. 3093–3118.

[32] G. Duan, B. Chen, X. Zhang, and Y. Wang. “A multiphase MPS solver for modeling multi-fluid interaction with free surface and its application in oil spill”. In: *Computer Methods in Applied Mechanics and Engineering* 320 (2017), pp. 133–161.

[33] G. Duan, S. Koshizuka, A. Yamaji, B. Chen, X. Li, and T. Tamai. “An accurate and stable multiphase moving particle semi-implicit method based on a corrective matrix for all particle interaction models”. In: *International Journal for Numerical Methods in Engineering* 115.10 (2018), pp. 1287–1314.

[34] N. Lanson and J. P. Vila. “Renormalized Meshfree Schemes I: Consistency, Stability, and Hybrid Methods for Conservation Laws”. In: *SIAM Journal on Numerical Analysis* 46.4 (2008), pp. 1912–1934.

[35] N. Lanson and J. P. Vila. “Renormalized Meshfree Schemes II: Convergence for Scalar Conservation Laws”. In: *SIAM Journal on Numerical Analysis* 46.4 (2008), pp. 1935–1964.

[36] J. Basic, N. Degiuli, and D. Ban. “A class of renormalised meshless Laplacians for boundary value problems”. In: *Journal of Computational Physics* 354 (2018), pp. 269–287.

[37] J. Basic, B. Blagojevic, M. Andrun, and N. Degiuli. “A Lagrangian Finite Difference Method for Sloshing: Simulations and Comparison with Experiments”. In: *Proceedings of the Twenty-ninth International Ocean and Polar Engineering Conference*. 2019, pp. 3412–3418.

[38] H. Takeda, S. M. Miyama, and M. Sekiya. “Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics”. In: *Progress of Theoretical Physics* 92.5 (Nov. 1994), pp. 939–960.

[39] J. P. Morris, P. J. Fox, and Y. Zhu. “Modeling low Reynolds number incompressible flows using SPH”. In: *Journal of Computational Physics* 136.1 (1997), pp. 214–226.

[40] A. J. C. Crespo, M. Gómez-Gesteira, and R. A. Dalrymple. “Boundary conditions generated by dynamic particles in SPH methods”. In: *Computers, Materials and Continua* 5.3 (2007), pp. 173–184.

[41] S. Adami, X. Y. Hu, and N. A. Adams. “A generalized wall boundary condition for smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 231.21 (2012), pp. 7057–7075.

[42] D. Violeau and B. D. Rogers. “Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future”. In: *Journal of Hydraulic Research* 54.1 (2016), pp. 1–26.

[43] L.D. Libersky and A.G. Petschek. “Smooth particle hydrodynamics with strength of materials”. In: *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. Berlin, Heidelberg: Springer, 1991, pp. 248–257.

[44] B. Bouscasse, A. Colagrossi, S. Marrone, and M. Antuono. “Nonlinear water wave interaction with floating bodies in SPH”. In: *Journal of Fluids and Structures* 42 (2013), pp. 112–129.

[45] Y. Chen, X. Zheng, S. Jin, and W. Duan. “A corrected solid boundary treatment method for Smoothed Particle Hydrodynamics”. In: *China Ocean Engineering* 31.2 (2017), pp. 238–247.

[46] A. Ferrari, M. Dumbser, E. F. Toro, and A. Armanini. “A new 3D parallel SPH scheme for free surface flows”. In: *Computers and Fluids* 38.6 (2009), pp. 1203–1217.

[47] R. Vacondio, B. D. Rogers, and P. K. Stansby. “Smoothed Particle Hydrodynamics: Approximate zero-consistent 2-D boundary conditions and still shallow-water tests”. In: *International Journal for Numerical Methods in Fluids* 69.1 (2012), pp. 226–253.

[48] G. Fourtakas, R. Vacondio, and B. D. Rogers. “On the approximate zeroth and first-order consistency in the presence of 2-D irregular boundaries in SPH obtained by the virtual boundary particle methods”. In: *International Journal for Numerical Methods in Fluids* 78.8 (2015), pp. 475–501.

[49] J. J. Monaghan. “Simulating free surface flows with SPH”. In: *Journal of Computational Physics* 110.2 (1994), pp. 399–406.

[50] J. J. Monaghan and A. Kos. “Scott Russell’s wave generator”. In: *Physics of Fluids* 12.3 (2000), pp. 622–630.

[51] J. J. Monaghan and J. B. Kajtar. “SPH particle boundary forces for arbitrary boundaries”. In: *Computer Physics Communications* 180.10 (2009), pp. 1811–1820.

[52] M. Ferrand, D. R. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis. “Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method”. In: *International Journal for Numerical Methods in Fluids* 71.4 (2013), pp. 446–472.

[53] S. Kulasegaram, J. Bonet, R. W. Lewis, and M. Profit. “A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications”. In: *Computational Mechanics* 33.4 (2004), pp. 316–325.

[54] A. Mayrhofer, M. Ferrand, C. Kassiotis, D. Violeau, and F. Morel. “Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D”. In: *Numerical Algorithms* 68.1 (2015), pp. 15–34.

[55] J. A. Feldman and J. Bonet. “Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems”. In: *International Journal for Numerical Methods in Engineering* 72.3 (2007), pp. 295–324.

[56] A. Mayrhofer, B. D. Rogers, D. Violeau, and M. Ferrand. “Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions”. In: *Computer Physics Communications* 184.11 (2013), pp. 2515–2527.

[57] A. Leroy, D. Violeau, M. Ferrand, and C. Kassiotis. “Unified semi-analytical wall boundary conditions applied to 2-D incompressible SPH”. In: *Journal of Computational Physics* 261 (2014), pp. 106–129.

[58] A. Leroy, D. Violeau, M. Ferrand, L. Fratter, and A. Joly. “A new open boundary formulation for incompressible SPH”. In: *Computers and Mathematics with Applications* 72.9 (2016), pp. 2417–2432.

[59] M. Ferrand, A. Joly, C. Kassiotis, D. Violeau, A. Leroy, F. Morel, and B. D. Rogers. “Unsteady open boundaries for SPH using semi-analytical conditions and Riemann solver in 2D”. In: *Computer Physics Communications* 210 (2017), pp. 29–44.

[60] E. S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. “Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method”. In: *Journal of Computational Physics* 227.18 (2008), pp. 8417–8436.

[61] R. H. Cole and R. Weller. “Underwater Explosions”. In: *Physics Today* 1 (1948), p. 35.

[62] R. A. Gingold and J. J. Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (1977), pp. 375–389.

[63] M. Basa, N. J. Quinlan, and M. Lastiwka. “Robustness and accuracy of SPH formulations for viscous flow”. In: *International Journal for Numerical Methods in Fluids* 60.10 (2009), pp. 1127–1148.

[64] R. Fatehi and M. T. Manzari. “A remedy for numerical oscillations in weakly compressible smoothed particle hydrodynamics”. In: *International Journal for Numerical Methods in Fluids* 67.9 (2011), pp. 1100–1114.

[65] R. Fatehi and M. T. Manzari. “A consistent and fast weakly compressible smoothed particle hydrodynamics with a new wall boundary condition”. In: *International Journal for Numerical Methods in Fluids* 68.7 (2012), pp. 905–921.

[66] J. P. Morris. “Simulating surface tension with smoothed particle hydrodynamics”. In: *International Journal for Numerical Methods in Fluids* 33.3 (2000), pp. 333–353.

[67] X.Y. Hu and N.A. Adams. “An incompressible multi-phase SPH method”. In: *Journal of Computational Physics* 227.1 (2007), pp. 264–278.

[68] S. Adami, X. Y. Hu, and N. A. Adams. “A transport-velocity formulation for smoothed particle hydrodynamics”. In: *Journal of Computational Physics* 241 (2013), pp. 292–307.

[69] C. Zhang, X. Y. Hu, and N. A. Adams. “A weakly compressible SPH method based on a low-dissipation Riemann solver”. In: *Journal of Computational Physics* 335 (2017), pp. 605–620.

[70] N. Govender, D. N. Wilke, S. Kok, and R. Els. “Development of a convex polyhedral discrete element simulation framework for NVIDIA Kepler based GPUs”. In: *Journal of Computational and Applied Mathematics* 270 (2014), pp. 386–400.

[71] N. Govender, P. Pizette, D. N. Wilke, and N. E. Abriak. “Validation of the GPU based blaze-DEM framework for hopper discharge”. In: *Proceedings of the 4th International Conference on Particle-Based Methods - Fundamentals and Applications, PARTICLES 2015*. 2015, pp. 81–92.

[72] D. N. Wilke, N. Govender, P. Pizette, and N. E. Abriak. “Computing with Non-convex Polyhedra on the GPU”. In: *Proceedings of the 7th International Conference on Discrete Element Methods*. Vol. 188. Springer Singapore, 2017, pp. 1371–1377.

[73] U. Ghia, K. N. Ghia, and C. T. Shin. “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method”. In: *Journal of Computational Physics* 48.3 (1982), pp. 387–411.

[74] S. Khorasanizade and J. M. M. Sousa. “A detailed study of lid-driven cavity flow at moderate Reynolds numbers using Incompressible SPH”. In: *International Journal for Numerical Methods in Fluids* 76.10 (2014), pp. 653–668.

[75] Z. Mao and G. R. Liu. “A Lagrangian gradient smoothing method for solid-flow problems using simplicial mesh”. In: *International Journal for Numerical Methods in Engineering* 113.5 (2018), pp. 858–890.

[76] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias. “Experimental investigation of dynamic pressure loads during dam break”. In: *Journal of Fluids and Structures* 48 (2014), pp. 407–434.