Computers and Fluids 190 (2019) 346–361

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel smoothed particle hydrodynamics (SPH) models

Georgios Fourtakas [a,*], Jose M. Dominguez [b], Renato Vacondio [c], Benedict D. Rogers [a]

[a] School of Mechanical, Aerospace and Civil Engineering, University of Manchester, Manchester M13 9PL, UK
[b] Environmental Physics Laboratory (EPHYSLAB), Universidade de Vigo, Spain
[c] Department of Civil and Environmental Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124, Parma, Italy

ARTICLE INFO

Article history:
Received 31 March 2018
Revised 29 May 2019
Accepted 5 June 2019
Available online 6 June 2019

Keywords:
Smoothed particle hydrodynamics
Wall boundary conditions
Density diffusion term correction
Local uniform stencil
Fictitious particles
Complex arbitrary geometries

ABSTRACT

This paper presents the development of a new boundary treatment for free-surface hydrodynamics using the smoothed particle hydrodynamics (SPH) method accelerated with a graphics processing unit (GPU). The new solid boundary formulation uses a local uniform stencil (LUST) of fictitious particles that surround and move with each fluid particle and are only activated when they are located inside a boundary. This addresses the issues currently affecting boundary conditions in SPH, namely the accuracy, robustness and applicability while being amenable to easy parallelization such as on a GPU. In 3-D, the methodology uses triangles to represent the geometry with a ray tracing procedure to identify when the LUST particles are activated. A new correction is proposed to the popular density diffusion term treatment to correct for pressure errors at the boundary. The methodology is applicable to complex arbitrary geometries without the need of special treatments for corners and curvature is presented. The paper presents the results from 2-D and 3-D Poiseuille flows showing convergence rates typical for weakly compressible SPH. Still water in a complex 3-D geometry with a pyramid demonstrates the robustness of the technique with excellent agreement for the pressure distributions. The method is finally applied to the SPHERIC benchmark of a dry-bed dam-break impacting an obstacle showing satisfactory agreement and convergence for a violent flow.

1. Introduction

Smoothed particle hydrodynamics (SPH) is becoming increasingly popular to apply to a range of applications including hydraulics, biomedical and nuclear applications [5,14,41–43]. However, the robust numerical simulation of these applications is highly dependent on the performance of the boundary conditions employed within the SPH model. Since the early application of SPH to free-surface flows and confined flows found throughout engineering applications, boundary conditions have been the subject of intense scrutiny and development [49]. Despite this concerted effort by the SPH community, imposing boundary conditions in SPH is still an open problem due to the Lagrangian nature of SPH and the kernel based interpolation. This is recognised by boundary conditions being identified as Grand Challenge in the Smoothed Particle Hydrodynamics European Research Interest Community (SPHERIC, http://www.spheric-sph.org).

Methodologies for imposing the solid wall boundary conditions can be grouped in three generic methodologies: repulsive functions, fictitious particles and boundary integrals [50].

The widely used repulsive function approach, proposed by Monaghan [35] and Monaghan and Kajtar [36] describes the walls by particles which exert a repulsive short-range force similar to a Leonard-Jones potential force on fluid particles. With this approach 2-D and 3-D irregular geometries can be easily discretized, but the kernel truncation near the wall is not explicitly treated and can introduce non-negligible inaccuracies.

Another widely used method to describe boundaries in SPH [1,7,27] is to use fictitious particles to represent the presence of the boundary. This can take the form of mirror or ghost particles as introduced by Randles and Libersky [39]. However, extending the method to 3-D is challenging for irregular geometries. Alternatively, fictitious particles can take the form of stationary fluid particles or similar [1,26] to whom appropriate properties are applied to enforce the physically correct conditions of impermeability. An example of such an approach is the dynamic boundary condition (DBC) [7] currently implemented in the DualSPHysics code [8,9] which is suitable to reproduce complex geometries and

* Corresponding author.
E-mail address: georgios.fourtakas@manchester.ac.uk (G. Fourtakas).

https://doi.org/10.1016/j.compfluid.2019.06.009

is ideally suited to parallelisation on heterogeneous architectures such as GPUs, but suffers from drawbacks such as over-dissipation and spurious pressure oscillation. The approach of Marrone et al. [27] which used interpolation of the particle properties inside the fluid domain transformed to the boundary particles reduces these effects, similar to the work of Adami et al. [1].

Another alternative is based on the work of Kulasegaram et al. [20] who proposed using an approximation to the surface integral to account for the truncated kernel. This introduces a correction factor into the SPH summations and additional terms in the conservation equations in order to mimic the presence of the boundary. The work of Kulasegaram et al. [20] uses an empirical function originating from variational principles to approximate the force. This concept was further developed in Bierbrauer et al. [4], De Leffe et al. [10], Marongiu et al. [25], Ferrand et al. [12] and Mayrhofer et al. [29] avoiding the use of an empirical function. These methods have the advantage of restoring zero consistency in the SPH interpolation and can reproduce to first order the physical conditions of Neumann wall boundary stress conditions and near wall shear stress for low Reynolds number flows. However, as noted recently by Valizadeh and Monaghan [49], the discretization of complex 3-D geometries and/or multi-phase flows is challenging [28]. More recently using a cut-face process and improved second-order operators, Chiron et al. [6] have extended the 3-D surface integral idea of Mayrhofer et al. [28] to the Riemann-based SPH formulation that employs an evolution equation for particle volumes.

None of the aforementioned approaches has emerged as being uniquely superior to other boundary techniques. Valizadeh and Monaghan [49] made a comparison of several boundary conditions showing that for weakly compressible SPH, the formulation of Adami et al. [1] is superior in terms of pressure fields in the vicinity of the wall and stability. However, it is not clear how this can be extended to 3-D geometries and eventually to higher order convergence. Whilst many of these boundary conditions work well for academic test cases with simple geometry, their extension to arbitrarily complex 3-D geometries is not straightforward. Furthermore, SPH is inherently computational expensive, therefore boundary conditions algorithms have to be suitable to be accelerated by means of heterogeneous hardware such as GPUs. This motivates the methodology developed in this paper.

An interesting variant of the fictitious particle approach was proposed by Ferrari et al. [13], who similar to Yildiz et al. [51], used a local point symmetry method which is able to discretize arbitrarily complex geometries without introducing empirical forces. This approach has some very clear theoretical attractions, namely that is it possible to generate an individual stencil for a particle of any phase or identity such as in multi-phase flows, and should be general to any geometry. However, when the method was applied to shallow water equations (SWE) [46] it was evident that the original proposal of Ferrari et al. [13] was insufficient to complete the missing kernel support and prevent particles from penetrating the boundary and an enhancement was needed. This was modified again for the Navier–Stokes equations in 2-D [16] to address issues in the inconsistencies in the moments of the kernel and their derivatives which are indicators of the accuracy of any boundary condition.

Although the virtual boundary particles methodology has merit when applied in 2-D, such an approach can be cumbersome in 3-D. Representation of 3-D domains using predefined virtual particles becomes computationally memory intensive since the 3-D domain boundary triangles need to be discretized with geometrical points. Moreover, each fluid particle interacts with all the virtual particles within its support and large numbers of virtual particles are required to represent the solid boundary in 3-D.

During the development work presented in Vacondio et al. [46] and Fourtakas et al. [16], it became apparent that another option existed to formulate a boundary condition that could fulfil all the attractions of the Ferrari et al. [13] method but in a practical manner. This idea is the focus of this paper which proposes a new formulation of the local fictitious particles approach where a local uniform stencil (LUST) follows each individual fluid particle and only fictitious particles located within the region of the boundary are activated. Boundary are represented by lines (in 2-D) or triangles (in 3-D) without introducing virtual particle discretization. The method is easy to extend to arbitrarily complex 3-D geometries and has been accelerated for heterogeneous architectures such as GPUs. Approximate zero-th and first order consistency are ensured by using a fully uniform fictitious particle stencil. The proposed wall boundary condition is implemented in the open-source GPU code DualSPHysics [9].

This paper is structured as follows; In Section 2, the governing equations and the discretization of the standard weakly compressible SPH equations is presented together with a correction for the density diffusion term. This is followed by the description of the methodology of the LUST fictitious particles and the implementation on GPU. The accuracy and robustness of the new methodology is then assessed with several validation cases including the Poiseuille flow in 2-D and 3-D, a 3-D still water case with a pyramid in the domain and the SPHERIC benchmark 3-D dam break impact test.

2. SPH formalism

In SPH, the kernel approximation of a scalar function f at position $\mathbf{x}$ in the continuum domain is obtained as:

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

where Ω is the volume of the integral, W is the smoothing kernel function, h is the smoothing length, used to define the influence area of the smoothing kernel function. In practical applications kernels have a compact support, meaning that W goes to zero at a finite distance from $\mathbf{x}$. In a discrete domain Eq. (1) can be approximated with a summation:

$$\langle f(\mathbf{x})\rangle = \sum_{j}^{N} f_j W(\mathbf{x}-\mathbf{x}_j, h)V_j, \tag{2}$$

where $f(\mathbf{x}_j)=f_j$ is the value of the scalar function f at particle j with position $\mathbf{x}_j$ and associated volume V_j and N is the total number of particles. The $\langle ...\rangle$ symbol denotes an SPH interpolation and will be dropped for simplicity in the rest of the paper.

Starting from Eq. (1) it is possible to derive the following Equation to approximate the derivative of gradient of a scalar function f:

$$\nabla f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\nabla W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}'. \tag{3}$$

The discrete approximation of Eq. (3) is the following:

$$\langle \nabla f(\mathbf{x})\rangle = \sum_{j}^{N} f_j \nabla W(\mathbf{x}-\mathbf{x}_j, h)V_j. \tag{4}$$

A variety of kernels have been proposed in the literature [23] such as the Gaussian kernel, B-spline kernels, and the 5th-order Wendland kernel. The latter has been adopted in this paper due to its simplicity and low computational requirements:

$$W(R) = \begin{cases} a_d\left(1-\frac{R}{2}\right)^4(2R+1) & 0 \le R \le 2 \\ 0 & R > 2 \end{cases}, \tag{5}$$

where $a_d = 7/(4\pi h)$ and $a_d = 21/(16\pi h)$, respectively, in a two- and three-dimensional space and $R = |\mathbf{x} - \mathbf{x}'|/h$. More details on the SPH formulation and recent developments can be found in [50].

3. SPH formulation

3.1. Governing equations

This section presents the governing equation in SPH form. The Navier–Stokes equations can be written in a continuous Lagrangian form for a weakly compressible fluid as

$$\begin{cases} \frac{d\rho}{dt} = -\rho\nabla\cdot\mathbf{u}, \\ \rho\frac{d\mathbf{u}}{dt} = -\nabla P + \nabla\cdot\boldsymbol{\tau} + \rho\mathbf{g}, \end{cases} \tag{6}$$

where $\mathbf{u}$ is the velocity, ρ is the density, P is the pressure, $\boldsymbol{\tau}$ is the deviatoric component of the total stress and $\mathbf{g}$ is the gravity acceleration.

The continuity and momentum equations are coupled by means of the Tait's equation of state (EOS):

$$P = \frac{c_0^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right], \tag{7}$$

with the polytrophic index $\gamma = 7$, $c_0 = \sqrt{\partial P/\partial\rho}$ is speed of sound and subscript 0 denotes reference values.

Herein we adopt the classical SPH formulation of Eq. (4) as the aim of the present paper is to present a novel way to discretize wall boundaries and not investigate more recent formulations now available in literature which address instabilities in SPH [22,38,44,47,48]. Without such stabilisation, the test cases presented in Section 6 are more challenging and might indicate possible problems in the boundary treatment.

Throughout this paper, i and j subscripts denote the interpolated particle and its neighbours, respectively. The density evolution and momentum of the particles follow the Navier–Stokes equations [33] using the addition of the δ-SPH formulation of Antuono et al. [3]

$$\frac{d\rho_i}{dt} = \sum_j^N m_j\mathbf{u}_{ij}\cdot\nabla W_{ij} + \delta h c_i\sum_j^N \psi_{ij}\cdot\nabla W_{ij}V_j, \tag{8}$$

$$\frac{d\mathbf{u}_i}{dt} = -\sum_{j=1}^N m_j\left(\frac{P_i+P_j}{\rho_i\rho_j} + \Pi_{ij}\right)\nabla W_{ij} + \mathbf{g}_i, \tag{9}$$

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_i, \tag{10}$$

where the subscripts ij denote the difference in value $f_{ij} = f_i - f_j$ for a field variable f, hence $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ and $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and m_j the mass of the neighbouring particle. In the dry-bed dam-break impacting an obstacle test case, the artificial viscosity [34]; Ferrari et al. [13] has been used to stabilize the solution and provide an artificial viscous force denoted as Π_{ij},

$$\Pi_{ij} = \begin{cases} \frac{-\alpha_\pi\bar{c}_{ij}}{\bar{\rho}_{ij}}\frac{h\mathbf{u}_{ij}\cdot\mathbf{x}_{ij}}{\mathbf{x}_{ij}^2} & \mathbf{u}_{ij}\cdot\mathbf{x}_{ij} < 0 \\ 0 & \mathbf{u}_{ij}\cdot\mathbf{x}_{ij} \geq 0 \end{cases}, \tag{11}$$

where α_π is the free parameter, and the overbar denotes the average values of the i and j particles such that $\bar{c}_{ij} = \frac{1}{2}(c_i + c_j)$ and $\bar{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j)$ are the average speed of sound and density, respectively.

All other test cases use the Laplacian operator as suggested by Morris et al. [37] to model the physical viscous stresses

$$\nabla\cdot\mu_i\nabla\mathbf{u}_i = \sum_1^N \frac{m_j(\mu_j+\mu_j)\mathbf{r}_{ij}\cdot\nabla W_{ij}}{\rho_j(r_{ij}^2+\eta^2)}\mathbf{u}_{ij}, \tag{12}$$

where $\eta = 0.001\,\mathrm{h}$ to avoid singularity as $r_{ij}\to 0$.

In the continuity equation the second term on the RHS is a diffusion term added to prevent spurious oscillations in the density field.

The time integration method adopted is the symplectic predictor-corrector scheme bounded by the CFL condition [35]. Further details on the time integration scheme can be found in Crespo et al. [9].

3.2. Improved density diffusion term for gravity driven flows

The diffusion term in Eq. (8), can take different forms accordingly to the formulation adopted for ψ_{ij} as clearly explained in Antuono et al. [2]. For example Molteni and Colagrossi [32] suggested that,

$$\psi_{ij} = 2\left(\rho_j - \rho_i\right)\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|}, \tag{13}$$

whereas later Antuono et al. [3] proposed the so-called δ-SPH formulation,

$$\psi_{ij} = \left(\rho_j - \rho_i - \frac{1}{2}\left(\nabla\rho_j^L + \nabla\rho_i^L\right)\cdot\mathbf{x}_{ij}\right)\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2}, \tag{14}$$

where $\nabla\rho_j^L$ and $\nabla\rho_i^L$ are the normalized gradients computed, respectively, at particle i and j. In comparison with Eq. (13), in Eq. (14) high order terms are introduced in Eq. (14) by Antuono et al. [3], to ensure consistency at the free surface. Therefore, δ-SPH formulation can be successfully adopted for gravity dominated flow (see Antuono et al. [2] and Green et al. [17] for a comprehensive analysis).

In the present work we introduce a different formulation for the diffusion term of Eq. (8), aiming at restoring consistency at the free surface avoiding to compute the density normalized gradients. The key idea is to use the same formulation proposed by Molteni and Colagrossi [32] but substituting the dynamic density with the total one. Thus the term ψ_{ij} takes the following form:

$$\psi_{ij} = 2\left(\rho_j^D - \rho_i^D\right)\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|}, \tag{15}$$

where the superscript D denotes the dynamic density or pressure. Since $\rho^D = \rho^T - \rho^H$ (where superscript T and H denote the total and hydrostatic component, respectively), Eq. (13) can be rewritten in term of the hydrostatic and total parts as

$$\psi_{ij} = 2\left(\rho_{ji}^{\mathrm{T}} - \rho_{ij}^{\mathrm{H}}\right)\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|}. \tag{16}$$

In the context of the weakly compressible SPH, the equation of state relates the density to the total pressure at the particle location. However, only the hydrostatic part of the pressure is needed. Using Eq. (7) the hydrostatic density difference can be obtained by

$$\rho_{ij}^H = \rho_0\left(\sqrt[\gamma]{\frac{P_{ij}^H + 1}{C_B}} - 1\right). \tag{17}$$

The term P_{ij}^{H} is simply the hydrostatic pressure difference of particle i and j,

$$P_{ij}^{\mathrm{H}} = \rho_0 g z_{ij}, \tag{18}$$

where z_{ij} is the vertical distance between particle i and j and

$$C_B = \frac{c_0^2\rho_0}{\gamma}. \tag{19}$$

In comparison with the formulation of the diffusive term proposed by Molteni and Colagrossi [32]. Eq. (16) improves the behaviour of pressure near the wall boundaries, as demonstrated in Section 6, avoiding to compute the normalized density gradient. However, it must be noted that the formulation of Antuono et al. [3] of Eq. (14) is general and it can be adopted for any type of flow,

whereas the use of total density in the diffusive term of Eq. (15) is expected to work better than Eq. (13) only for gravity-dominated flows.

4. Wall boundary conditions methodology

4.1. Wall representation using triangles

The representation of the solid boundary line using SPH particles is well documented in literature [49,50]. SPH particles located on the boundary are either used as interpolation points or for geometrical purposes. With the former approach the governing equations are approximated on the boundary particles whereas with the latter approach line points serve the function of geometrical points for the generation of a set of fictitious particles within the truncated kernel support of the solid boundary.

The representation of the solid wall boundary using line points or virtual boundary particles has gained popularity recently [13,16,46]. The main advantage of this approach derives from the ease that fictitious particles can be generated at *runtime* without the need to predefine fictitious particles within the solid wall. Furthermore, the ability to readily generate a local uniform stencil within the solid boundary is beneficial satisfying the consistency criteria associated with the kernel support resulting on zeroth and first order consistency for uniform fluid domains. Fig. 1 illustrates the main mechanism of the virtual boundary particles methodology as proposed by Fourtakas et al. [16].

Herein, a different approach is proposed to represent the solid boundary line but retaining the aforementioned advantages of the virtual boundary particles methodology. In the proposed method, boundaries are discretized by means of triangles (in 3-D) without introducing virtual boundary particles. The triangulated surfaces can be readily used in 3-D without special treatments when discretizing arbitrary complex geometries. In this approach the discretization of the solid boundary line is independent of the particle resolution of the domain. In the absence of virtual particles, the fully uniform fictitious stencil is translated according to the position of the fluid particle as shown in Fig. 2. Each fluid particle interacts only with triangles located within its support by using the ray casting algorithm [40] reducing the interaction drastically when generating fictitious particles. The mechanism used to complete the truncated support near the boundary using the triangles is described next.

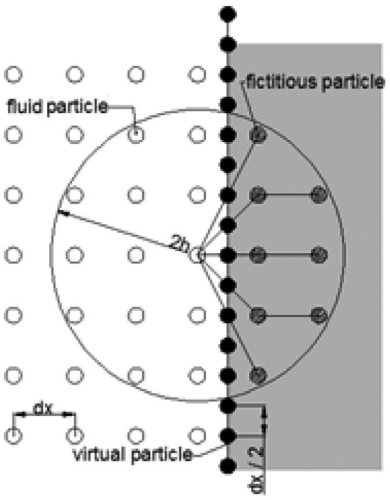


Fig. 1. Local uniform stencil generation using virtual boundary particles in 2-D [16].

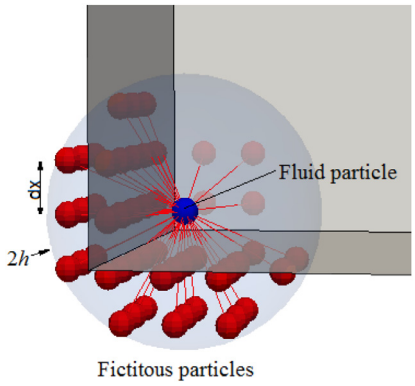


Fig. 2. Local uniform stencil generation using triangulated surfaces in 3-D. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.).

4.2. Local uniform stencil (LUST)

The idea of a local point of symmetry fictitious particle generation mechanism is not new to SPH. In Ferrari et al. [13], Vacondio et al. [46] and Fourtakas et al. [16] virtual particles are used to generate a set of uniform fictitious particles to complete the truncated kernel support depending on the distance of the fluid particle to the solid boundary and thus maintain zero-th and first-order consistency for uniform particle resolutions. The authors have demonstrated that the zero-th and first-order consistency for non-uniform particle resolutions is approximately satisfied.

In contrast to the work of Fourtakas et al. [16] and Vacondio et al. [46] where fictitious particles are generated depending on the distance of the fluid particle to the solid boundary, in the new scheme proposed herein, the use of virtual particles is replaced with a local uniform stencil that surrounds each fluid particle. The unique stencil is generated for an arbitrary particle at the beginning of the simulation and stored in memory.

When the support of a fluid particle is truncated by a solid wall represented by triangles, fictitious particles within the LUST stencil located within the fluid domain are discarded and only fictitious particles located within the solid wall actively contribute to the summations in Eqs. (8) and (9). Fig. 2 demonstrates a 3-D example where a fluid particle (shown in blue) is located near a corner. The particles shown in dark red are the fictitious particles belonging to the LUST stencil that are located inside the boundary and will therefore contribute to the summations as described below.

The main difference with the previous methods is that the distance from the fluid particle to the fictitious particle is constant and depends on the particle spacing Δx and kernel characteristics only, resulting in a uniform particle distribution within the solid wall. An example of a stencil generated for a fluid particle located at a distance $2\Delta x > d > \Delta x$ and $\Delta x > d > 0$ is shown in Fig. 3. Note that, due to the particle regular distribution, the method is able to guarantee that the zeroth- and first-order moments are equal to 1 and 0, respectively. Thus, the theoretical rate of convergence can be achieved for the SPH operators [21]. Lind and Stansby [21] and later Fourtakas et al. [15] showed that maintaining a regular particle distribution improves the accuracy and the convergence rate of SPH interpolation. Thus, herein we maintain the uniform distribution near the boundary and non-uniform distribution of fluid particles within the domain due to the Lagrangian nature of the formulation.

Each fictitious particle must be given values of particle properties in order to impose boundary conditions for the velocity and pressure of the fluid at the boundary. To ensure mass conservation and satisfy Eq. (8), the mass of the fictitious particle is equal to the

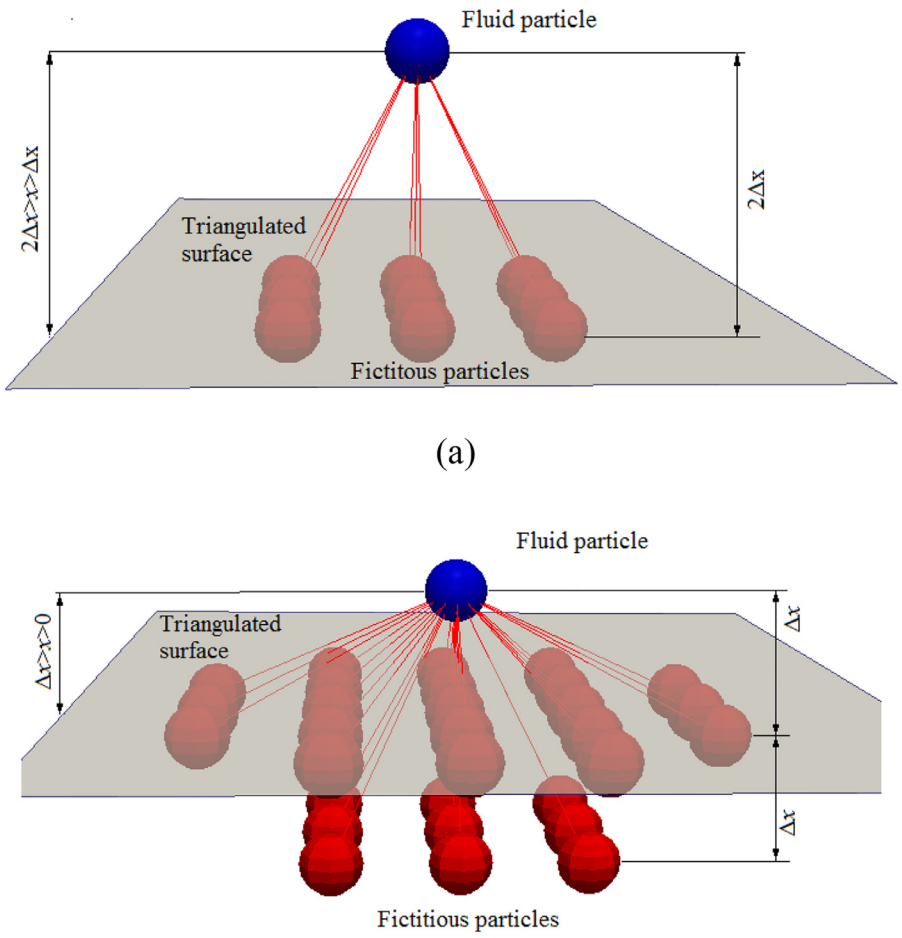


Fig. 3. Fluid particle support generation for a particle located at a distance (a) $2\Delta x > x > \Delta x$ and (b) $\Delta x > x > 0$ away from the boundary surface.

mass of the interior particle.

$$m_f = m_i, \tag{20}$$

where the subscript f refers to the fictitious particle.

To impose no-slip boundary conditions the velocity of the fictitious particles $\mathbf{u}_f$ is assigned according to Takeda et al. [45] method by

$$\mathbf{u}_f = (\mathbf{u}_i - \mathbf{u}_v)\frac{r_{vf}}{r_{iv}} - \mathbf{u}_v, \tag{21}$$

where the subscript v denotes the virtual wall, r_{vf} is the perpendicular distance between the fictitious particle and the boundary triangle according to $r_{vf} = \mathbf{r}_{vf} \bullet \mathbf{n}$ where n is the normal to the surface pointing into the fluid. Similarly, r_{iv} is the perpendicular distance between the fluid particle i and the boundary triangle as shown in Fig. 4 and $\mathbf{u}_f$ and $\mathbf{u}_v$ is fluid particle velocity and the physical wall velocity, respectively.

A similar formulation to Eq. (20) is used for the pressure of the fictitious particles with the added correction for the hydrostatic pressure

$$P_f = P_i + \rho_0 \mathbf{g} \cdot \mathbf{r}_{if}\frac{r_{vf}}{r_{iv}}, \tag{22}$$

which ensures that $\partial P/\partial \mathbf{n} \rightarrow \infty$ as the particle distance $r_{iv} \rightarrow 0$ from the solid wall is going to zero, imposing a no-penetration boundary condition on the solid wall by a repulsive pressure.

Finally, the density of the fictitious particles is a function of the pressure of Eq. (21) according to the equation of state Eq. (7)

$$\rho_f = \rho_0 \sqrt[\gamma]{\frac{P_f}{C_B} + 1}. \tag{23}$$

In Table 1 the pseudocode of the LUST algorithm is shown. For each particle interaction the LUST stencil is moved accordingly to the position of the fluid particle array `pa[i]` (line 2), then the summation over all active fictitious particles of the stencil starts with the loop at line 3. For each jth fictitious particle of the stencil the ray casting algorithm is used to assess whether it is inside or outside the fluid region. If the jth particle is outside the fluid region, then physical quantities are assigned to the fictitious particle (line 13) by Eqs. (19)–(22) and finally the contributions to the summation of Eqs. (8) and (9) of particle j are computed (line 14).

It should be noted that within this work the boundary treatment avoids the need to correct the kernel or gradient to account for the missing kernel support. The applicability of the methodology is demonstrated for both 2-D and 3-D geometries.

5. Implementation on GPU

Recently, the emergence of graphic processing units (GPUs) for scientific computing has enabled acceleration of massively data-parallel computations. The architecture of GPUs is particularly suitable for SPH simulations as an energy efficient and cost effective

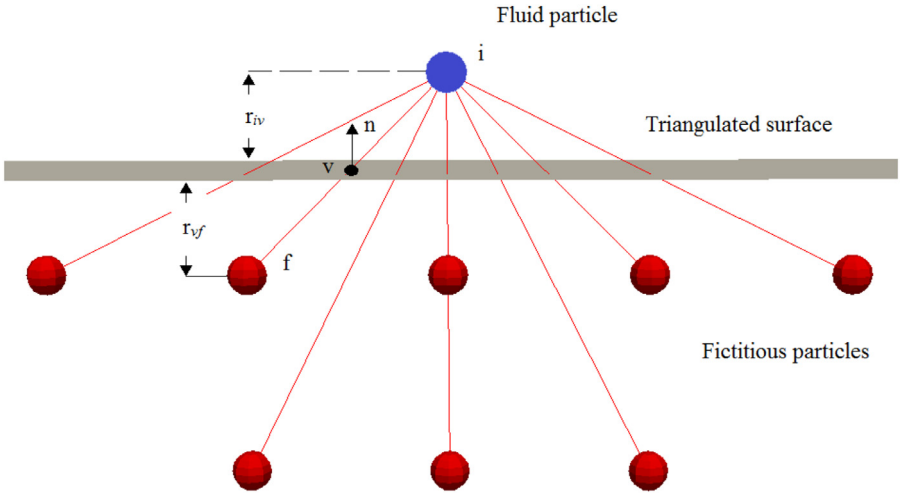


Fig. 4. Fluid particle support generation and normal distances from the fluid particle to the virtual wall and fictitious particle.

Table 1
Pseudocode of the LUST algorithm.

```
//loop over all fluid particles
1: for i=1, n_p
2:   call move_LUST_stencil(…)
     //loop over all fictitious particles of the LUST stencil
3:   for j=1, n_fic
     //Ray casting algorithm
4:     foreach side in polygon:
       //p(j) is the j_th fictitious particle
5:     if ray_intersects_segment(pa[j],side) then
6:       count ← count+1
7:     if is_odd(count) then
8:       return inside
9:     else
10:      return outside
11:    end if
       // fictitious particles outside the fluid region
12:      if (outside) then
         //compute mass, density and velocity for fictitious particle j
13:      call compute_physical_quantities(pa[i], pa[j])
         //compute particle interaction between particle i and j
14:      call particle_interaction(pa[i],pa[j])
15:    end if
16:  end for
17: end for
```

Table 2
Parameters of the SPH model for the 2-D Poiseuille flow configuration.

Parameter	Value
Time integration	Predictor corrector
SPH Kernel	Wendland
Density diffusion	No
Courant number	0.2
c_{s0}	10 m/s
$h/\ \Delta x$	2.0
Viscosity model	Morris operator
	$\mu = 0.1$ Pa s
Density	$\rho = 1$ kg/m^3

option [24]. The attraction comes from the parallel, multithreaded many-core computing architecture of NVIDIA GPU cards which is well suited for HPC applications in general. Different SPH codes have been parallelized to exploit the computational power of GPUs [9,18].

The GPU memory is organized in several different types that can be used by programmers to achieve high Computation to Global Memory Access (CGMA) ratio and thus high efficiency of the solver. Variables which are stored in registers and in shared memory can be accessed with low latency in a highly parallel manner. While registers are private to each thread, shared memory can be accessed by all threads that belongs to the same block. However, registers and shared memory sizes are limited, ranging from a few bytes to kilobytes, respectively. On the other hand the global memory is "off-chip" with slower communication. Finally, the constant memory allows read-only access by the device and provides faster and more parallel data access paths than the global memory. Therefore, appropriate implementation of the numerical model can have a big effect on the speedup achieved.

As described in the Section 4, each fluid particle has a predefined stencil of fictitious particles. This predefined stencil moves with the fluid particle throughout the simulation but does not change during the simulation so it is created and stored in the global GPU memory only once at the beginning. When computing acceleration for a fluid particle located close to the wall, a test should be performed to determine which fictitious particles in the predefined stencil are located in the boundary region. To determine whether a fictitious particle lies in the boundary region, it is necessary to check if the line segment connecting the fluid particle to the fictitious one intersects any of the triangles that define the wall. If so, the triangle with the intersection point closest to the fluid particle is chosen. In any case, the ray casting algorithm [40] in 3-D is used to determine if the fictitious particle is valid, whether it lies in the boundary region or not.

The number of triangles required to define complex geometries can be significantly high for fine resolutions. In order to reduce the number of triangles included in the test, the neighbour-list algorithm [11] used for the simulation is altered to include a list of triangle neighbours in each neighbour-list cell. The list of triangles in each cell is created and stored in the GPU memory at the beginning of the simulation. However, the list must be updated if the boundary position undergoes displacement. Achieving an efficient GPU implementation for the LUST algorithm is a complex task due to multiple loops in the code and memory accesses required to determine the fictitious particles. An option to increase the performance is to store the relevant triangle information in shared memory but the limiting factor is the restricted size of the shared memory per multiprocessor (less than 64 Kbytes for compute capability 6.0 or higher for NVIDIA GPUs) referred to as "OneStep".

Another option to increase the GPU efficiency is by splitting the force computation summation into two different CUDA kernels (referred to as "TwoStep") for each fluid particle. In the first CUDA

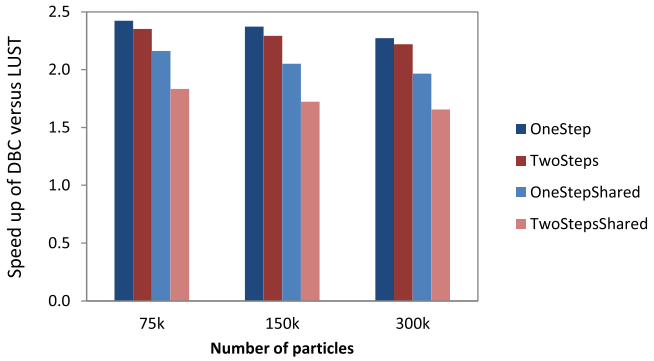


Fig. 5. Speed up of DBC over LUST boundary wall boundary conditions.

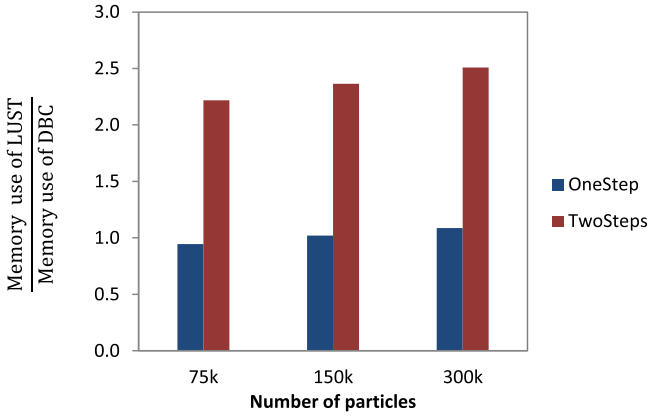


Fig. 6. Increasing factor in GPU memory compared to DBC.

Table 3
Parameters of the SPH model for the 3-D Poiseuille flow configuration.

Parameter	Value
Time integration	Predictor-corrector
SPH Kernel	Wendland
Density diffusion	No
Courant number	0.2
c_{s0}	3 m/s
$h/\ \Delta x$	2.0
Viscosity model	Morris operator
	$\mu = 10^{-4}$ Pa s
Density	$\rho = 1$ kg/m^3

Table 4
Parameters of the SPH model for the still water in 3-D including a pyramid configuration.

Parameter	Value
Time integration	Predictor-corrector
SPH Kernel	Wendland
Density diffusion	0.1
Courant number	0.2
c_{s0}	58 m/s
$h/\ \Delta x$	2.0
Viscosity model	Morris operator
	$\mu = 0.01$ Pa s

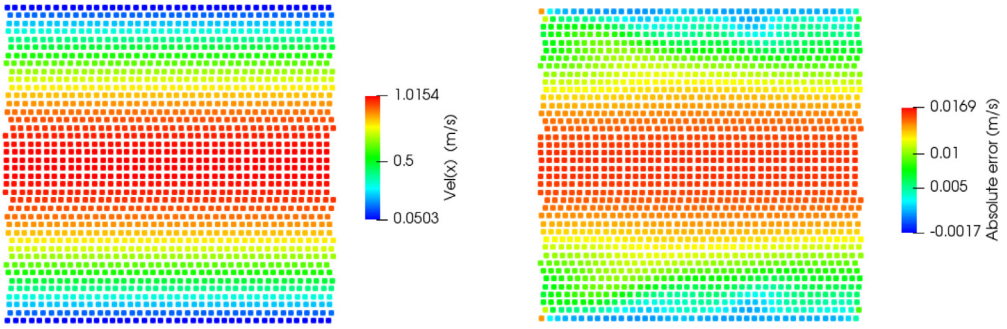
kernel, the boundary triangle that intersects the segment joining the fluid and the fictitious particle is defined (if present). The interaction with the fluid particle is then performed using the second CUDA kernel. This process significantly reduces the code complexity, decreases the register occupancy and minimizes irregular memory access. Also, a combination of the "TwoStep" algorithm with the shared memory is possible. In the latter case, the GPU memory increases since the triangle for each point of the predefined stencil needs to be stored in the global memory for every fluid particle (referred to as "OneStepShared", "TwoStepShared"). Results for performance and memory usage have been analysed using a 3-D dam-break impact with obstacle test case (SPHERIC benchmark test #2) presented in Section 6.3. The results are compared with the DBC [7,9] currently available in the open-source DualSPHysics in Fig. 5 using an NVIDIA Tesla K20c GPU card. The DBC is faster than the new approach and the speedup is as seen in Fig. 5. Nevertheless, results obtained in Section 6 with LUST in comparison to the DBC for the 3-D dam break show better agreement with the experimental data. On the other hand, the LUST boundary condition is also more memory-consuming than DBC, shown in Fig. 6. Figs. 5 and 6 present results for the first version ("OneStep"), with the improvement when using two steps ("TwoStep") and with the use of shared memory ("OneStepShared", "TwoStepShared").

6. Test cases

6.1. Poiseuille

6.1.1. Poiseuille flow in 2-D

To test the accuracy and convergence characteristics of the LUST boundary conditions in the absence of gravity and its ability to impose a non-slip boundary condition, a laminar Poiseuille flow is



(a) Velocity component along axial direction.

(b) Velocity component along axial direction absolute error.

Fig. 7. 2-D Poiseuille flow: velocity field at 15 s.

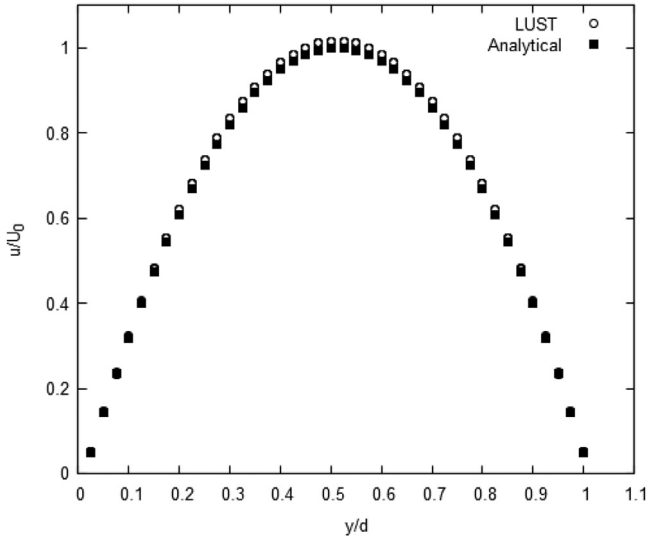


Fig. 8. 2-D normalised velocity field comparison with the analytical solution.

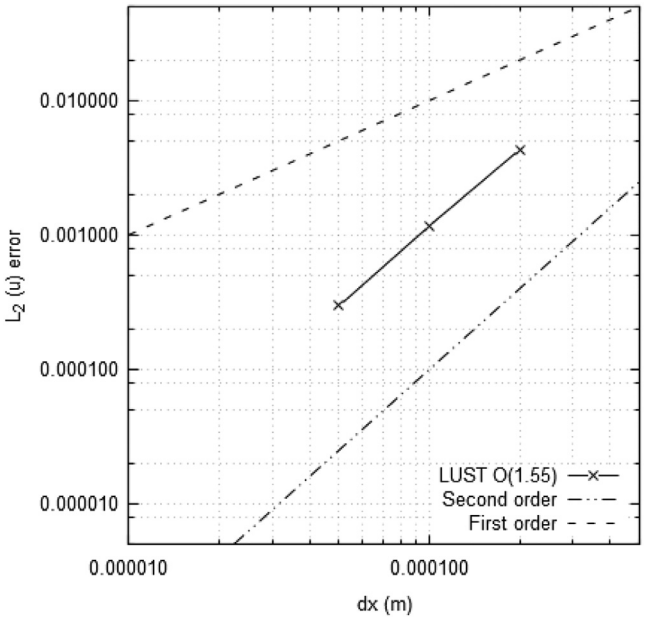


Fig. 9. L_2 error norm of the velocity at $t = 10$ s and $Re = 10$.

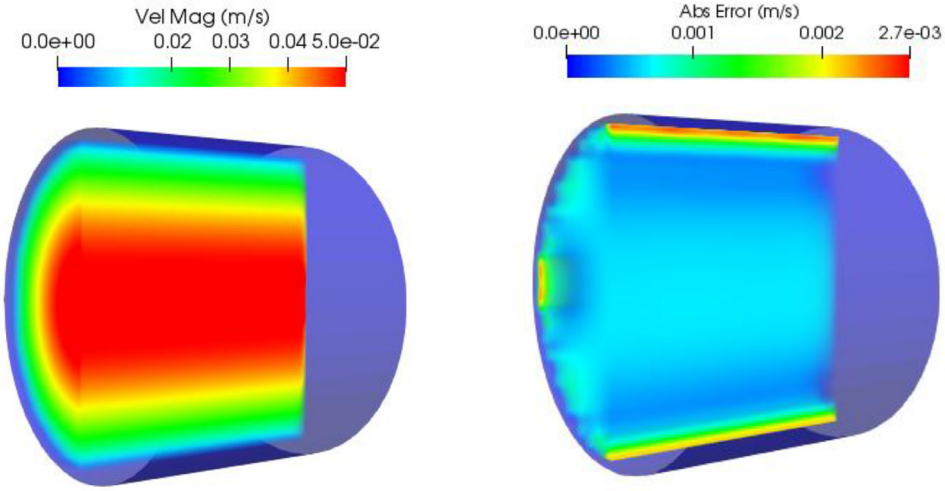
simulated. Initially, a 2-D case is performed with a Reynolds number $Re = 10$ using open periodic boundaries. The channel width was set to $d = 1$ m with a freestream velocity of $U_\infty = 1.0$ m/s. For this test case, to obtain meaningful results the formulation of Morris et al. [37] is used to model the viscous forces. The particle resolution for this test case is $\Delta x = 0.2$, 0.1 and 0.05 m resulting in 110, 420 and 1640 particles, respectively. The same case has been simulated by numerous researchers including Ferrand et al. [12]. The adopted models and parameters for the SPH model are shown in Table 2.

Fig. 7 shows the velocity field for the test case with $\Delta x = 0.05$ m after steady state has been reached at $t = 15$ s along with the absolute velocity error. It is notable that the majority of the absolute error is not close to the wall boundaries but instead in the interior fluid domain. Note, that the SPH formulation that has been used in this test case is the classical SPH formulation [34] without the use of density diffusion so that any errors near the wall boundary are evident which would not be visible otherwise by using a diffusion term or other density filtering techniques.

Fig. 8 shows a comparison of the velocity profile at the middle of the domain against the analytical solution. Similarly, the majority of inaccuracy against the analytical solution is within the fluid domain and not the wall boundaries. The L_2 norm of the velocity error of the 2-D Poiseuille flow are shown in Fig. 9 with a satisfactory order of convergence of 1.55.

6.1.2. Poiseuille flow in 3-D

Although the 2-D Poiseuille flow demonstrated good accuracy and satisfactory convergence, the purpose of the LUST methodology is its ability to handle arbitrary 3-D geometries such as sharp corners as demonstrated in the 3-D still water test cases below and curved boundaries. Therefore, the Hagen–Poiseuille 3-D flow which requires a 3-D tube in the absence of gravity is an ideal test case. The Reynolds number was set to $Re = 5$, with a channel diameter of $d = 1 \times 10^{-2}$ m with a freestream velocity of $U_\infty = 5 \times 10^{-2}$ m/s. The particle resolutions for this test case are $\Delta x = 0.001$, 0.0005 and



(a) Magnitude of velocity field. (b) Magnitude of velocity field absolute error.

Fig. 10. 3-D Hagen–Poiseuille flow.

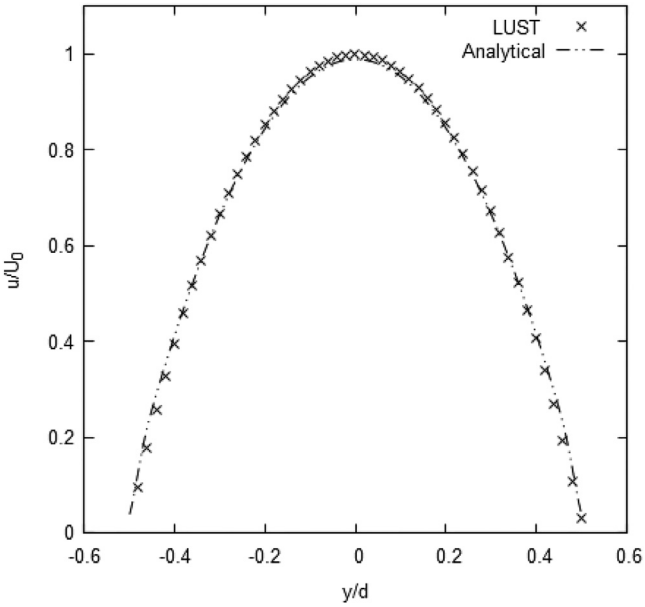


Fig. 11. Velocity profile comparison with the analytical solution of the Hagen–Poiseuille flow.

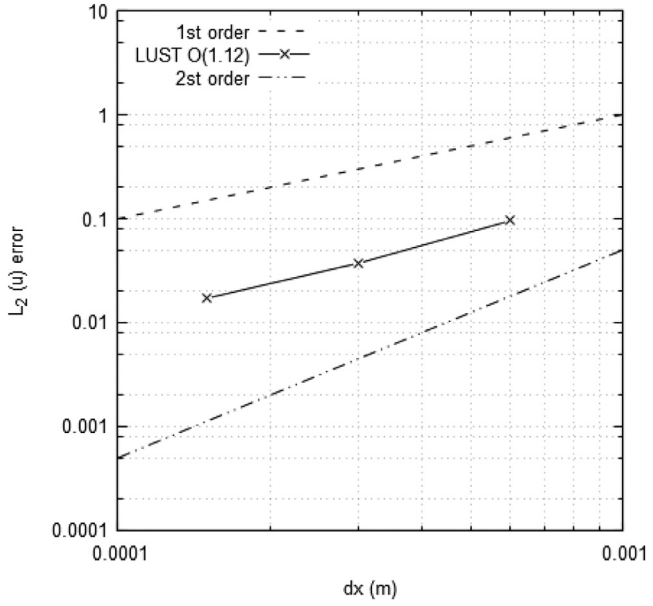
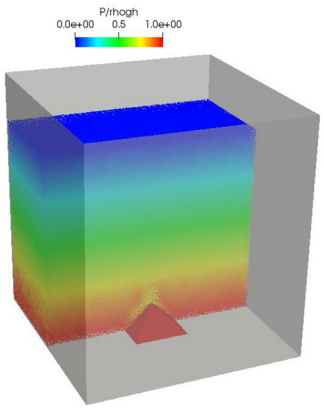
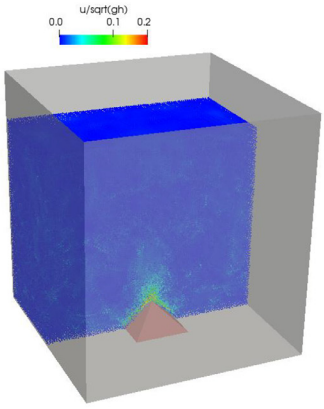


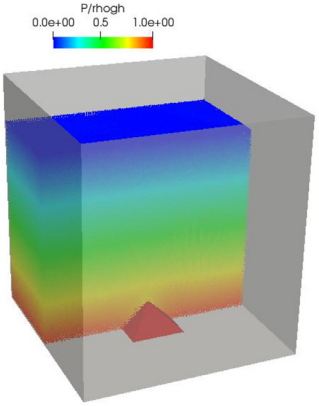
Fig. 12. L_2 error norm of the velocity at $t = 15$ s and $Re = 5$ for the 3-D Hagen–Poiseuille flow. The order of convergence is approximately 1.12.



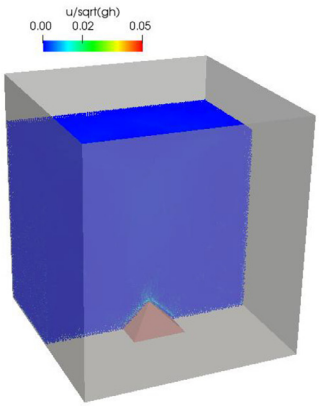
(a) Normalised pressure field without corrected density diffusion term.



(b) Normalised magnitude of velocity field without corrected density diffusion term.

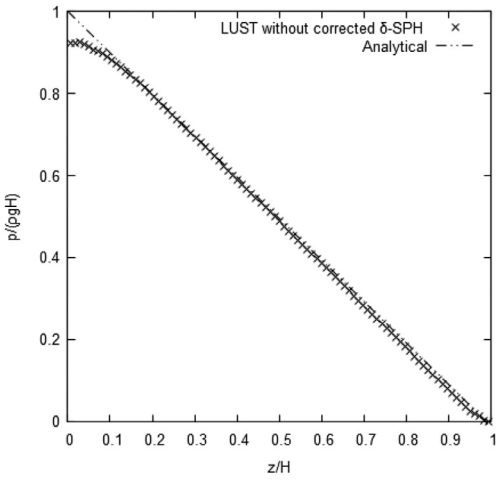


(c) Normalised pressure field with corrected density diffusion term.

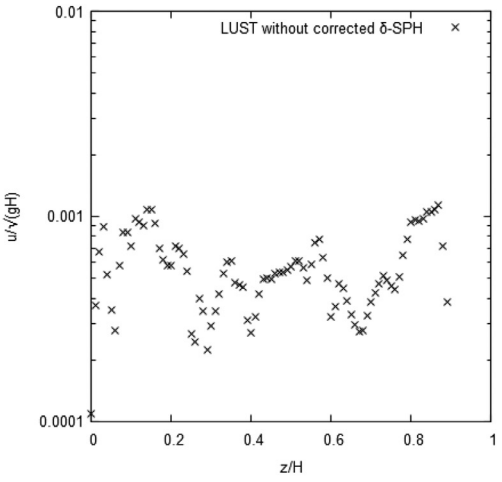


(d) Normalised magnitude of velocity field with corrected density diffusion term.

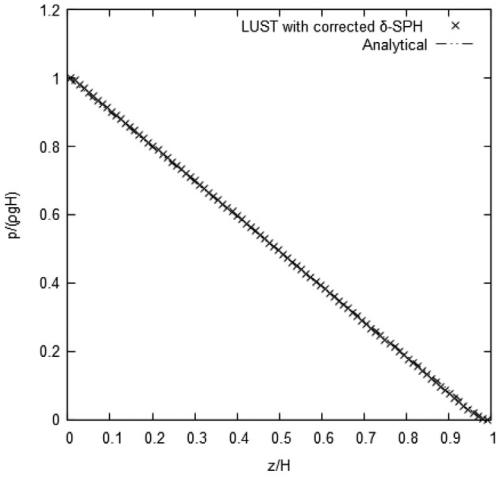
Fig. 13. 3-D still water with pyramid at 10 s comparison of the normalised pressure and velocity field magnitude without and with the corrected density diffusion term for $\Delta x = 0.001$ m. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article).



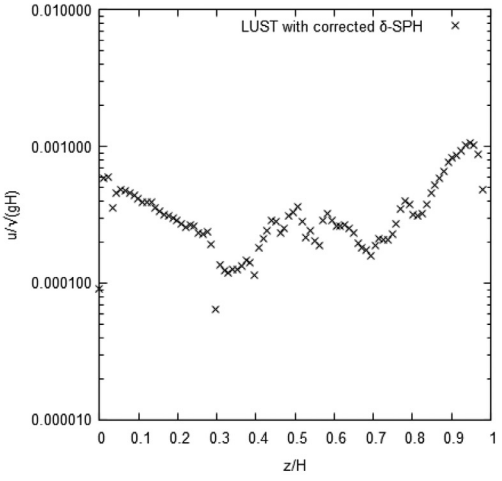
(a) Normalised pressure profile, uncorrected density diffusion term.



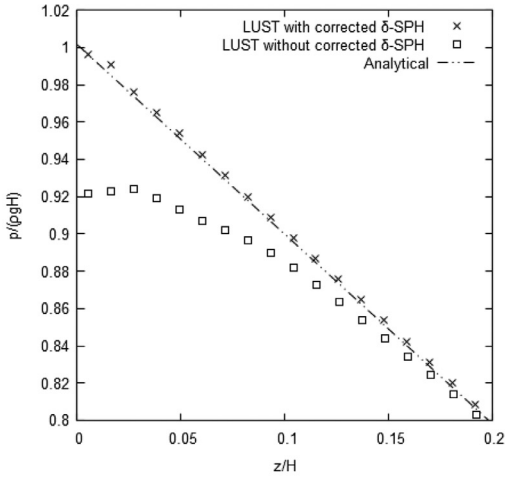
(b) Normalised velocity profile, uncorrected density diffusion term.



(c) Normalised pressure profile with corrected density diffusion term.

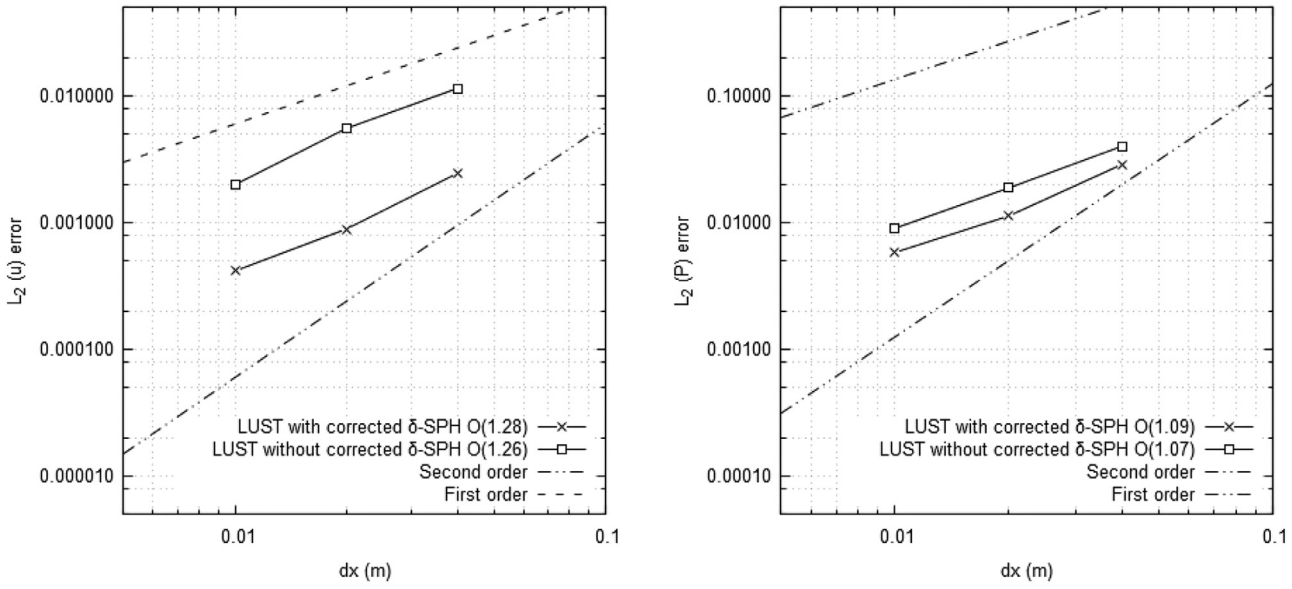


(d) Normalised velocity profile with corrected density diffusion term.



(e) Normalised pressure profile zoom-in near the wall boundary comparison between uncorrected and corrected density diffusion term for $\Delta x = 0.001$ m.

Fig. 14. 3-D still water with pyramid: profiles at $(x,y) = L/\sqrt{2}$ at 10 s for uncorrected and corrected density diffusion term.



(a) L_2 error norm, velocity (u). (b) L_2 error norm, pressure (p).

Fig. 15. 3-D still water with pyramid: L_2 error norms with the GPU solver using the Morris operator for the viscosity calculations at 10 s for uncorrected and corrected density diffusion term.

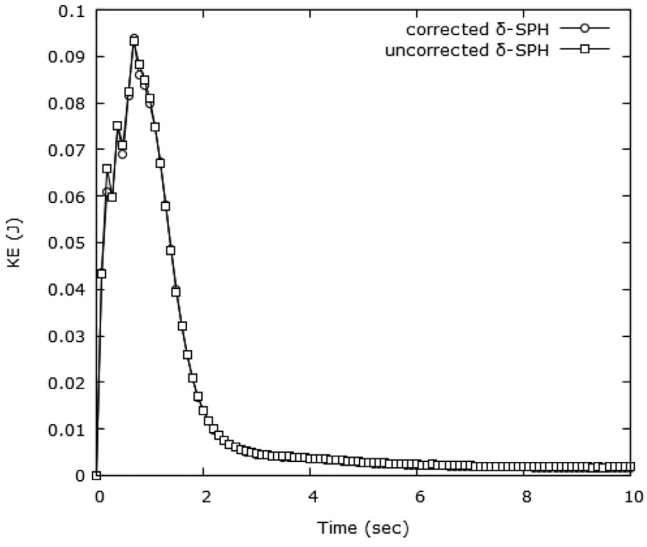


Fig. 16. Kinetic energy evolution time for the 3-D still water with pyramid.

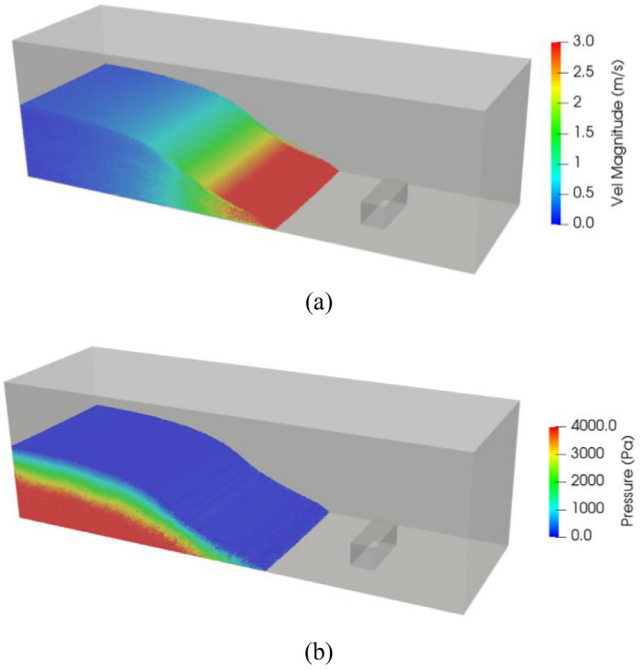


Fig. 17. 3-D dam break (a) velocity and (b) pressure field at $t = 0.3$ s.

0.00025 m resulting in 549, 5263 and 46,215 particles, respectively. The test case configuration of the SPH model is shown Table 3.

A cross section of the velocity field of the laminar Hagen–Poiseuille flow with a Reynolds number of $Re = 5$ is shown in Fig. 10 for the simulation with $\Delta x = 0.00025$ m. The flow field near the boundaries and within the domain is smooth without any spurious velocities. A plot of the absolute error of the velocity magnitude is shown in Fig. 10(b) with less than 4% absolute error to the freestream velocity. Also, a comparison of the velocity profile with the analytical solution with satisfactory results is shown in Fig. 11 followed by the convergence behaviour at steady state at time $t = 15$ s with an order of convergence of 1.12 shown in Fig. 12.

6.2. Still water in 3-D including a pyramid

To evaluate the performance of the proposed methodologies in presence of free-surface and gravity driven flow, the still water has been simulated in a 3-D square box with a pyramid at the bottom of the square tank. The pyramid was inserted to demonstrate the ability of the methodology to deal with more complex geometries, such as discontinuous points with internal and external angles and slopes within the computational domain. The classical SPH weakly compressible approach of Monaghan [34] with the corrected and uncorrected density diffusion term of Section 3. B. has been used together with the formulation of Morris et al. [37] for the viscous term with $\mu = 0.01$ Pa s.

The dimensions of the container box are $1 \times 1 \times 1.2$ m and a square base pyramid of sides 0.25 m with a height of 0.125 m

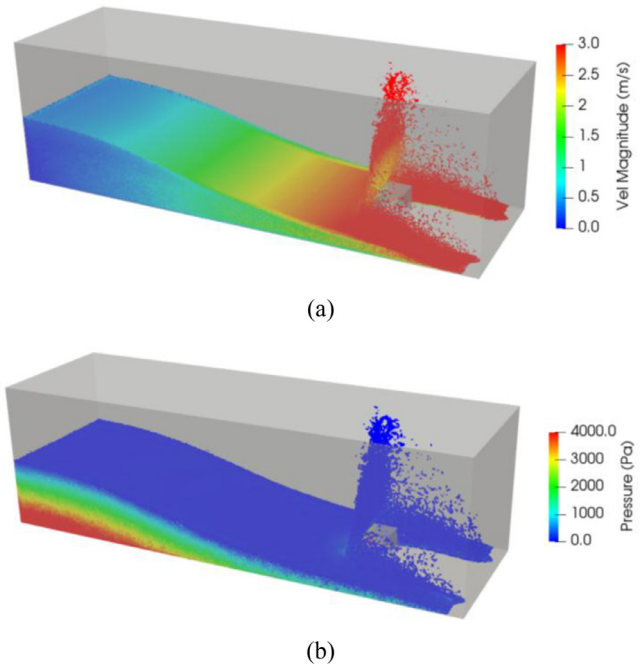


Fig. 18. 3-D dam break (a) velocity and (b) pressure field at $t = 0.6$ s.

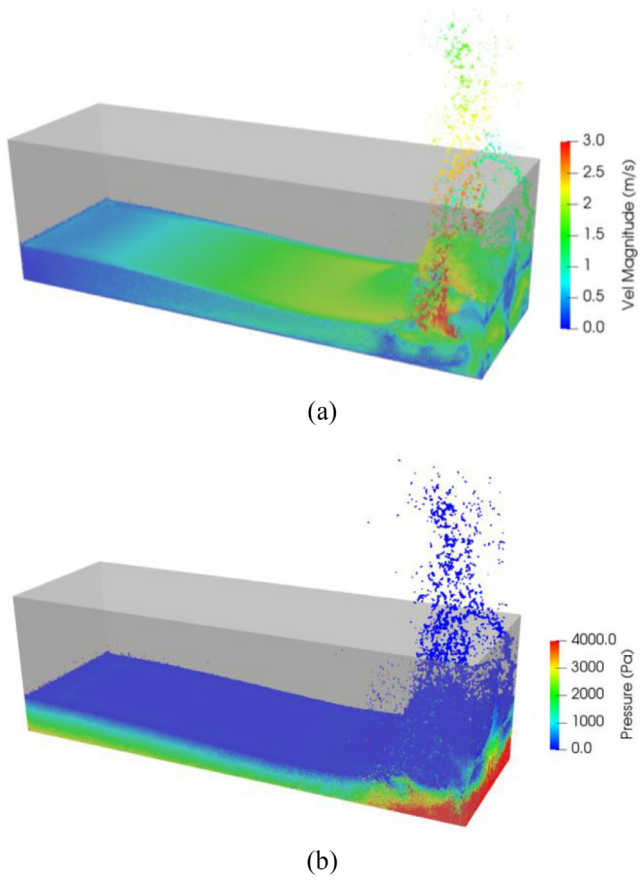


Fig. 19. 3-D dam break (a) velocity and (b) pressure field at $t = 1.2$ s.

and π / 2 radians angle is centered at the middle of the domain at $\boldsymbol{x} = (0.5, 0.5, 0.5)$ m. The water level elevation in the container is $H = 1$ m with an initial water density of $\rho_0 = 1000\,\text{kg/m}^3$. Only gravity acts on the fluid with $g = -9.81\,\text{m/s}^2$. The particle resolutions for this test case are $\Delta x = 0.04$, 0.02 and 0.001 m resulting in 12,061, 105,358 and 869,450 particles, respectively. Table 4 lists the parameters adopted for the still water in 3-D including a pyramid.

Fig. 13 shows the velocity and pressure field of the domain cross section at $x = 0.5$ m using the original formulation of the density diffusion term (Eq. (13)) and the correction proposed in Section 3.2. based on dynamic density (Eq. (15)). The pyramid is shown in red color in both images. Evidently results obtained with the proposed correction are in agreement with the analytical solution and the velocity magnitude is reduced by an order of magnitude.

The non-dimensional pressure and velocity profiles at $(x,y) = L/\sqrt{2}$ of the domain are shown in Fig. 14 for $\Delta x = 0.001$ m. Clearly, in Fig. 14(a) the uncorrected density diffusion term shows a dip in the pressure near the wall boundary on the order of 10% of the total pressure that is eliminated in Fig. 14(c) of Eq. (14). A comparison near the wall is provided in Fig. 14(e) by zooming into the region near the wall boundary where the error of the uncorrected density diffusion term is clearly visible. Similarly, the fluctuations in velocity shown in Fig. 14. (d) Are reduced by the use of the new density diffusion term correction.

The convergence study on the velocity and pressure L_2 errors is shown in Fig. 15 with the reasonable order of convergence of approximately 1.3 and 1.1, respectively, for the velocity and pressure. Note that, although the order of convergence has improved marginally by the corrected density diffusion term, the reduction in error between the two density diffusion term formulations is significant on the order of one magnitude.

In addition to the pressure and velocity field and convergence study, Fig. 16 shows the variation of the total kinetic energy of the fluid particles with time for $\Delta x = 0.001$ m . Evidently, after the initial settling of the fluid due to the weakly compressible formulation, the total kinetic energy decays rapidly. No significant difference between the corrected and uncorrected density diffusion term diffusion was observed.

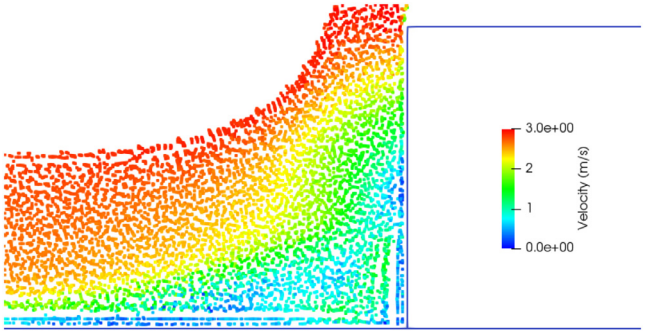


Fig. 20. 3-D dam break close-up near the obstacle for velocity $t = 0.6$ s.

Table 5
Parameters of the SPH model for the still water in 3-D including a pyramid configuration.

Parameter	Value
Time integration	Predictor-corrector
SPH Kernel	Wendland
Density diffusion	0.1
Courant number	0.2
c_{s0}	46 m/s
$h/\ \Delta x$	1.3
Viscosity model	Artificial viscosity $\alpha = 0.001$

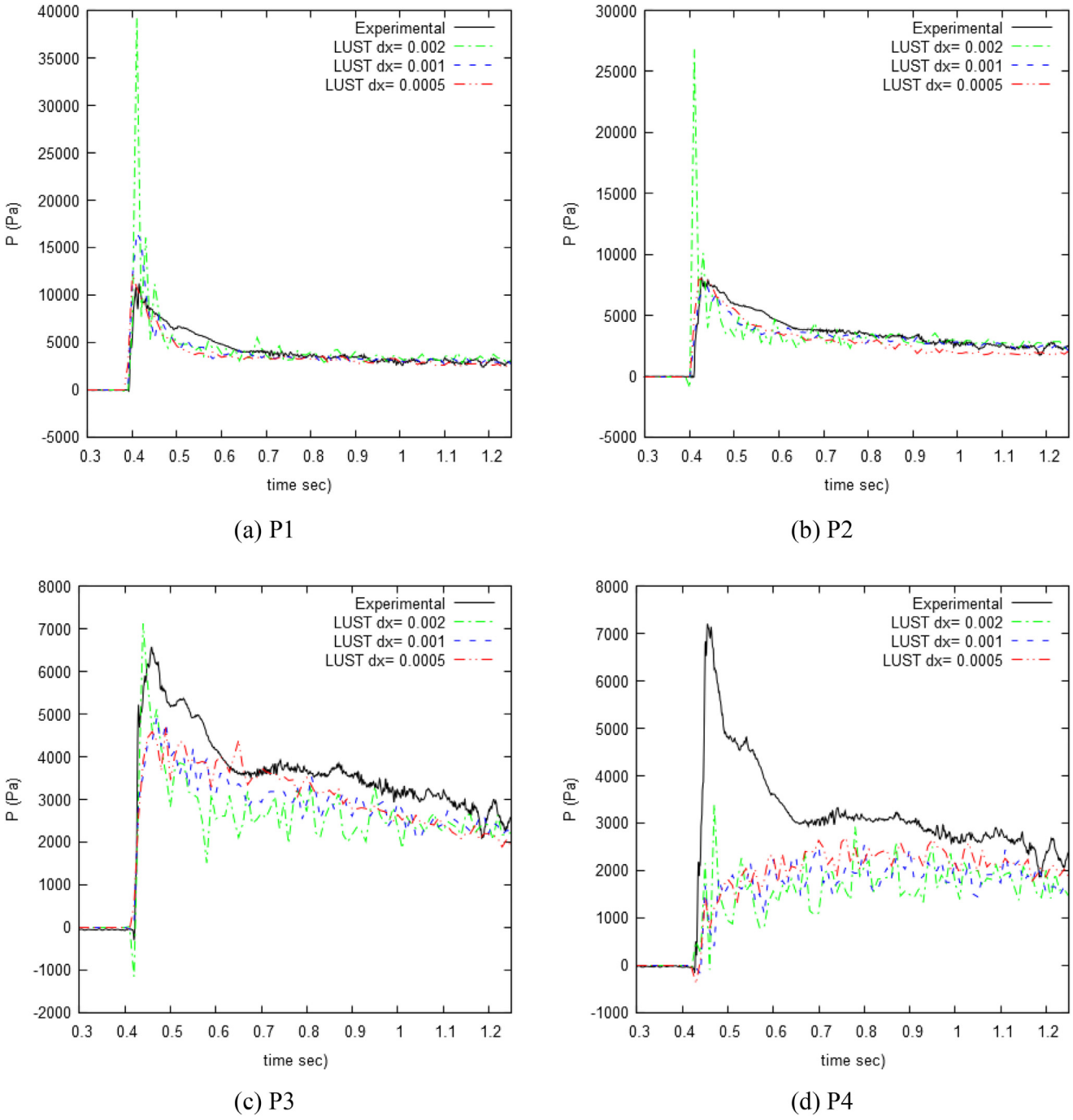
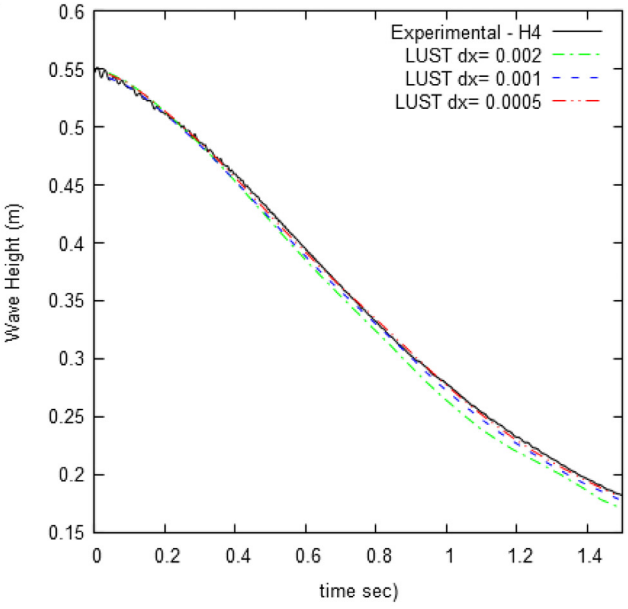
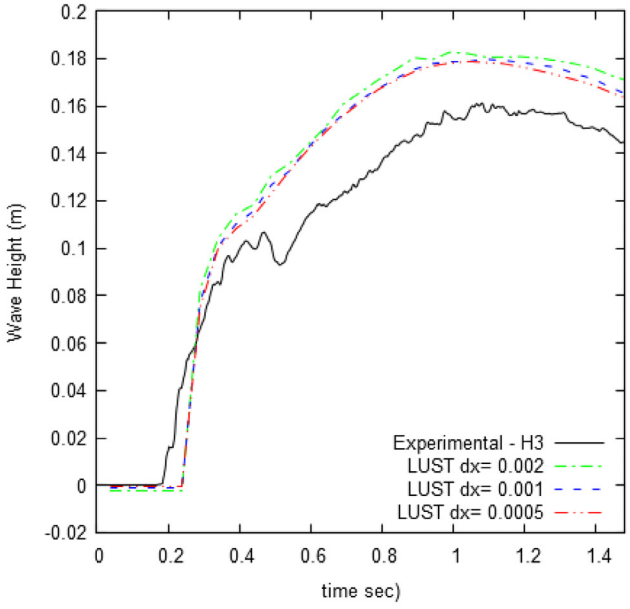


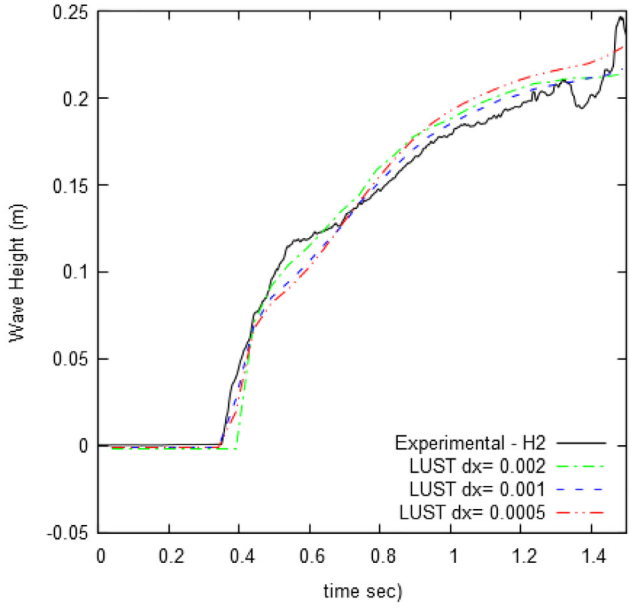
Fig. 21. 3-D dam break comparison with experimental data for pressures.



H4



H3



H2

Fig. 22. 3-D dam break comparison with experimental data for water height.

Table 6
Computational times for the 3-D Poiseuille, still water and dam break test cases, showing the cost per time step and particle per time step for all tested resolutions.

Case	Resolution (m)	No. of particles	Particle computational time per step (sec)	Computational time per time step (sec)
Poiseuille flow in 3-D	0.001	549	4.52E−05	0.044
	0.0005	5263	3.31E−05	0.238
	0.00025	46,215	1.50E−05	0.818
Still water in 3-D	0.04	12,061	4.14E−07	0.005
	0.02	105,358	1.61E−07	0.017
	0.01	869,450	1.16E−07	0.101
Dam break 3-D	0.002	83,692	1.91E−07	0.016
	0.001	669,735	1.32E−07	0.089
	0.0005	5,384,940	4.25E−08	0.229

6.3. 3-D dam break (SPHERIC Benchmark Test Case #2)

Although the still water test case is ideal to demonstrate the ability of the LUST to deal with hydrostatic conditions and the Poiseuille flow shows the noise of the domain and no-slip characteristics imposed on the wall boundaries, SPH is ideal to simulate applications with for non-linear behaviour such as high-speed impact flows and fragmentation in the presence of a free surface. Thus, the SPHERIC benchmark #2 has been chosen to demonstrate the ability of SPH in conjunction with the LUST wall boundaries to deal with such demanding flows.

The test case involves a breaking dam flow that further impacts a structure downstream where pressure and water height gauges have been placed. The reader is directed to the SPHERIC benchmark #2 test case [19], for the geometrical of the configuration and pressure and water height probes. The particle resolutions for this test case are $\Delta x = 0.002$, 0.001 and 0.0005 m. Table 5 lists the parameters adopted for the still water for the SPHERIC benchmark #2. In this case, we have used $h/\Delta x = 1.3$ is standard practice, particularly for computationally expensive 3-D simulations using WC-SPH such as used in DualSPHysics.

Fig. 17 shows the velocity field of the domain before impact, Fig. 18 at the impact and Fig. 19 after the impact of the breaking dam on the structure. Evidently, the velocity field is smooth and no fluctuations are present in the fluid domain or near the boundaries. However, more interesting are the flow features near the structure impacted by the breaking dam. Fig. 20 shows a slight separation of the near-boundary particles on the order of Δx / 2. This is due to the greater velocity of particles arriving from the left flowing over particles with near zero velocity immediately adjacent to the boundary.

Without resolving the boundary layer (which would require a computationally prohibitive resolution) a simulation with a finer resolution would mitigate this effect. This is another advantage of the LUST methodology by imposing a non-permeable wall boundary with non-slip characteristics. Finally, a comparison of the pressure against the experimental results is shown in Fig. 21 and Fig. 22 shows the water height probes against the experimental results for all three different particle resolutions. The pressures and water heights are well predicted especially as the resolution of the particle spacing increases which shows convergence to the experimental results with closer agreement to advanced multi-phase solver of Mokos et al. [30] which highlights the improvements over previous wall boundary conditions used in DualSPHysics [9]. Results in Fig. 21(d) are consistent with other authors [9] for probe P4 located on the top of the obstacle where the effects of air affect the experimental and numerical results.

To demonstrate the simulation runtimes, the computational times for the 3-D cases are shown in Table 6 for the tested resolutions. By refining the particle resolution the computational cost of a particle per time step is reducing and the computational time per time step is increasing with expected rate. For the 3-D dam break which uses sufficient number of particles (5.3 million) for the high resolution, scalability of the computational cost per time step is evident without reaching a plateau.

7. Conclusions and discussion

This paper has presented a new boundary treatment for free-surface hydrodynamics for SPH using a local uniform stencil (LUST) of fictitious particles that surround and move with each fluid particle. The LUST particles are only activated when they are located inside a boundary. The methodology employs a ray tracing procedure with triangles representing the geometry to identify when the LUST particles are activated. The new solid boundary formulation addresses the issues currently affecting boundary conditions in SPH, namely the accuracy, robustness and applicability and is straightforward to parallelize such as a GPU demonstrated here. A new correction to the density diffusion term treatment corrects for pressure errors at the boundary showing much closer agreement than standard δ-SPH for hydrostatic pressure distributions for the challenging case of still water in a complex 3-D geometry with a pyramid. The methodology is applicable to arbitrary complex geometries without the need of special treatments for corners and curvature. Results from 2-D and 3-D Poiseuille flows shows that the method converges demonstrating the robustness of the technique with excellent agreement for the velocity profiles. The method is finally applied to the SPHERIC benchmark of a dry-bed dam-break impacting an obstacle showing satisfactory agreement and convergence for a violent flow.

The method now stands to open the route forward as a robust and easy-to-implement boundary treatment which will be of great potential benefit to more complex SPH schemes such as variable particle resolution [47] and multi-phase flows [30,31]. Since the local uniform stencil is generated using the fluid's smoothing length h the generated support can in take any shape required by the p-adaptivity scheme (tetrahedral, hexahedral etc.). The resulting fictitious support can be generated locally for any h without the need of further kernel corrections. A variable resolution methodology has not been investigated here as it is the focus of future research.

Acknowledgements

This work has been funded by the EPSRC, UK grant number EP/L014890/1, by the Ministry of Education, Universities and Research (Italian government) under the Scientific Independence of young Researchers project, grant number RBSI14R1GP, CUP code, by Xunta de Galicia (Spain) under project ED431C 2017/64 "Programa de Consolidación e Estructuración de Unidades de Investigación Competitivas (Grupos de Referencia Competitiva)" co-funded by European Regional Development Fund (FEDER). The

work is also funded by the Ministry of Economy and Competitiveness of the Government of Spain under project "WELCOME ENE2016-75074-C2-1-R".

References

[1] Adami S, Hu X, Adams N. A generalized wall boundary condition for smoothed particle hydrodynamics. J Comput Phys 2012;231(21):7057–75.
[2] Antuono M, Colagrossi A, Marrone S. Numerical diffusive terms in weakly–compressible SPH schemes. Comput Phys Commun 2012;183(12):2570–80.
[3] Antuono M, Colagrossi A, Marrone S, Molteni D. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. Comput Phys Commun 2010;181(3):532–49.
[4] Bierbrauer F, Bollada P, Phillips T. A consistent reflected image particle approach to the treatment of boundary conditions in smoothed particle hydrodynamics. Comput Method Appl Mech Eng 2009;198(41):3400–10.
[5] Canelas R, Domínguez J, Crespo A, Gómez-Gesteira M, Ferreira R. Resolved simulation of a granular-fluid flow with a coupled SPH-DCDEM model. J Hydraul Eng 2017;143(9):06017012.
[6] Chiron L, de Leffe M, Oger G, Le Touzé D. Fast and accurate SPH modelling of 3D complex wall boundaries in viscous and non viscous flows. Comput Phys Commun 2019;234:93–111.
[7] Crespo A, Gómez-Gesteira M, Dalrymple RA. Boundary conditions generated by dynamic particles in SPH methods. CMC-Tech Sci Press 2007;5(3):173.
[8] Crespo AC, Dominguez JM, Barreiro A, Gomez-Gesteira M, Rogers BD. GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods. PLoS One 2011;6(6):e20685.
[9] Crespo AJC, Domínguez JM, Rogers BD, Gómez-Gesteira M, Longshaw S, Canelas R, Vacondio R, Barreiro A, García-Feal O. DualSPHysics: open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). Comput Phys Commun 2015;187:204–16.
[10] De Leffe M, Le Touzé D, Alessandrini B. Normal flux method at the boundary for SPH. In: Proceedings of the fourth international SPHERIC workshop (SPHERIC 2009); 2009.
[11] Domínguez JM, Crespo AJ, Gómez Gesteira M, Marongiu JC. Neighbour lists in smoothed particle hydrodynamics. Int J Numer Methods Fluids 2011;67(12):2026–42.
[12] Ferrand M, Laurence D, Rogers BD, Violeau D, Kassiotis C. Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method. Int J Numer Methods Fluids 2013;71(4):446–72.
[13] Ferrari A, Dumbser M, Toro EF, Armanini A. A new 3D parallel SPH scheme for free surface flows. Comput Fluids 2009;38(6):1203–17.
[14] Fourtakas G, Rogers B. Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using Smoothed Particle Hydrodynamics (SPH) accelerated with a Graphics Processing Unit (GPU). Adv Water Res 2016;92:186–99.
[15] Fourtakas G, Stansby PK, Rogers BD, Lind SJ. An Eulerian–Lagrangian incompressible SPH formulation (ELI-SPH) connected with a sharp interface. Comput Method Appl Mech Eng 2018;329:532–52.
[16] Fourtakas G, Vacondio R, Rogers BD. 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. Int J Numer Methods Fluids 2015;78(8):475–501.
[17] Green MD, Vacondio R, Peiró J. A smoothed particle hydrodynamics numerical scheme with a consistent diffusion term for the continuity equation. Comput Fluids 2019;179:632–44.
[18] Hérault A, Bilotta G, Dalrymple RA. SPH on GPU with cuda. J Hydraul Res 2010;48(S1):74–9.
[19] Kleefsman K, Fekken G, Veldman A, Iwanowski B, Buchner B. A volume-of-fluid based simulation method for wave impact problems. J Comput Phys 2005;206(1):363–93.
[20] Kulasegaram S, Bonet J, Lewis R, Profit M. A variational formulation based contact algorithm for rigid boundaries in two-dimensional SPH applications. Comput Mech 2004;33(4):316–25.
[21] Lind S, Stansby P. High-order Eulerian incompressible smoothed particle hydrodynamics with transition to Lagrangian free-surface motion. J Comput Phys 2016;326:290–311.
[22] Lind S, Xu R, Stansby P, Rogers BD. Incompressible smoothed particle hydrodynamics for free-surface flows: a generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. J Comput Phys 2012;231(4):1499–523.
[23] Liu M, Liu G, Lam K. Constructing smoothing functions in smoothed particle hydrodynamics with applications. J Comput Appl Math 2003;155(2):263–84.
[24] Longshaw SM, Rogers BD. Automotive fuel cell sloshing under temporally and spatially varying high acceleration using GPU-based Smoothed Particle Hydrodynamics (SPH). Adv Eng Software 2015;83:31–44.
[25] Marongiu J-C, Leboeuf F, Caro J, Parkinson E. Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method. J Hydraul Res 2010;48(sup1):40–9.
[26] Marrone S, Antuono M, Colagrossi A, Colicchio G, Graziani G. Enhanced boundary treatment in 2D smoothed particle hydrodynamics models. In: Proceedings of the XIX Congress. AIMETA; 2009.
[27] Marrone S, Antuono M, Colagrossi A, Colicchio G, Le Touzé D, Graziani G. δ-SPH model for simulating violent impact flows. Comput Meth Appl Mech Eng 2011;200(13):1526–42.
[28] Mayrhofer A, Ferrand M, Kassiotis C, Violeau D, Morel F-X. Unified semi-analytical wall boundary conditions in SPH: analytical extension to 3-D. Numer Algorithms 2015;68(1):15–34.
[29] Mayrhofer A, Rogers BD, Violeau D, Ferrand M. Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. Comput Phys Commun 2013;184(11):2515–27.
[30] Mokos A, Rogers BD, Stansby PK. A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. J Hydraul Res 2017;55(2):143–62.
[31] Mokos A, Rogers BD, Stansby PK, Domínguez JM. Multi-phase SPH modelling of violent hydrodynamics on GPUs. Comput Phys Commun 2015;196:304–16.
[32] Molteni D, Colagrossi A. A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH. Comput Phys Commun 2009;180(6):861–72.
[33] Monaghan J. On the problem of penetration in particle methods. J Comput Phys 1989;82(1):1–15.
[34] Monaghan JJ. Smoothed particle hydrodynamics. Annu Rev Astron Astrophys 1992;30(1):543–74.
[35] Monaghan JJ. Simulating free surface flows with SPH. J Comput Phys 1994;110(2):399–406.
[36] Monaghan JJ, Kajtar JB. SPH particle boundary forces for arbitrary boundaries. Comput Phys Commun 2009;180(10):1811–20.
[37] Morris JP, Fox PJ, Zhu Y. Modeling low Reynolds number incompressible flows using SPH. J Comput Phys 1997;136(1):214–26.
[38] Oger G, Marrone S, Le Touzé D, De Leffe M. SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms. J Comput Phys 2016;313:76–98.
[39] Randles P, Libersky L. Smoothed particle hydrodynamics: some recent improvements and applications. Comput Method Appl Mech Eng 1996;139(1–4):375–408.
[40] Roth SD. Ray casting for modeling solids. Comput Graph Image Process 1982;18(2):109–44.
[41] Schnabel D, Özkaya E, Biermann D, Eberhard P. Modeling the motion of the cooling lubricant in drilling processes using the finite volume and the smoothed particle hydrodynamics methods. Comput Method Appl Mech Eng 2018;329:369–95.
[42] Souto-Iglesias A, Delorme L, Pérez-Rojas L, Abril-Pérez S. Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics. Ocean Eng 2006;33(11–12):1462–84.
[43] Sun P, Colagrossi A, Marrone S, Antuono M, Zhang A. Multi-resolution Delta–plus-SPH with tensile instability control: towards high Reynolds number flows. Comput Phys Commun 2017;224:63–80.
[44] Sun PN, Colagrossi A, Marrone S, Antuono M, Zhang AM. Multi-resolution Delta-plus-SPH with tensile instability control: towards high Reynolds number flows. Comput Phys Commun 2017;224:63–80.
[45] Takeda H, Miyama SM, Sekiya M. Numerical simulation of viscous flow by smoothed particle hydrodynamics. Prog Theor Phys 1994;92(5):939–60.
[46] Vacondio R, Rogers B, Stansby P. Smoothed Particle Hydrodynamics: approximate zero-consistent 2-D boundary conditions and still shallow-water tests. Int J Numer Methods Fluids 2012;69(1):226–53.
[47] Vacondio R, Rogers BD, Stansby PK, Mignosa P. Variable resolution for SPH in three dimensions: towards optimal splitting and coalescing for dynamic adaptivity. Comput Method Appl Mech Eng 2016;300:442–60.
[48] Vacondio R, Rogers BD, Stansby PK, Mignosa P, Feldman J. Variable resolution for SPH: a dynamic particle coalescing and splitting scheme. Comput Method Appl Mech Eng 2013;256:132–48.
[49] Valizadeh A, Monaghan JJ. A study of solid wall models for weakly compressible SPH. J Comput Phys 2015;300:5–19.
[50] Violeau D, Rogers BD. Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. J Hydraul Res 2016;54(1):1–26.
[51] Yildiz M, Rook R, Suleman A. SPH with the multiple boundary tangent method. Int J Numer Methods Eng 2009;77(10):1416–38.