PHYSICAL REFLECTIVE BOUNDARY CONDITIONS IN PARTICLE METHODS: A COLLISION DETECTION AND RESPONSE ALGORITHM

Carlos Alberto Dutra Fraga Filho

cadff1@gmail.com

Development, Implementation and Application of Computational Tools for Problem Solving in Engineering Research Group, Federal Institute of Espírito Santo, Mechanical Engineering Coordination, Avenida Vitória, 1729, Zip Code: 29040-620, Jucutuquara, Vitória– ES, Brazil.

Abstract. *This paper presents the implementation of a computational algorithm, based on the physical laws of motion and foundations of analytic geometry, for applying of physical reflective boundary conditions in particle methods. The computational tool is capable of detecting the particle collisions against the boundaries and to carry out the necessary reflections, at each numerical iteration, in order to keep them inside the domain. The contribution of this research is in the step-by-step description of the algorithm implemented, including case studies. Numerical simulations have been carried out for hydrostatic and hydrodynamic problems. The simulations results are in agreement with analytical and experimental results. The computational algorithm implemented showed excellent results in the studied cases.*

Keywords: *physical reflective boundary conditions, collision detection and response, SPH particle method, kinetic energy restitution coefficient, computational algorithm*

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

1 INTRODUCTION

An algorithm for the treatment of boundaries in particle methods, employing fundamentals of physical particle reflections for collision detection and response, has been implemented. The motivation of this work was is the desire to fill the existing gap in the literature dedicated to the presentation of this reflexive boundary condition, based on the laws of motion and the fundamentals of analytic geometry.

Particles are assumed to be smooth, rigid and spherical, with non-zero masses, and defined by their volumes and fluid density. The impact model is an analytical approximation in which only the normal component of the impulse is considered. The periods of interaction between particles and contours are negligibly small instants of time.

Newton's laws of motion and the restitution law are obeyed. The reflections of physical particles immediately after impact against the solid plane are based on the fundamentals of physics and analytical geometry.

When a fluid is flowing, in addition to the loss of energy due to friction between the layers of the fluid (the viscosity effect), there is also a loss of energy due to the contact of the fluid with the walls of the reservoir, channel or pipes. The coefficient of restitution of energy (CR) is employed to measure the loss of kinetic energy in the collisions of particles with a rigid contour.

Models for collision treatment in particle methods aim to define how the particles interact with the contours, which prevent direct contact with the environment. Descriptions of these models can be found in (House & Keyser, 2007; Kelager, 2006).

The physical contours were defined using perpendicular geometric planes that defined a box (a 2D domain). However, there are no limitations on the implementation of the various planes that intersect at angles different of 90º. The collisions and trajectories of the particles colliding with the solid walls have been studied, after the solution of the momentum balance equation and the updating of the centre of mass and velocity of each particle using temporal integration.

In the first stage, the walls delimiting the domain are not considered. The particles move without physical limitations (obstacles within the space) within the time step defined for a single numerical iteration. In the second stage, the detection of collisions between each particle and the planes is employed. Immediately after collision detection, the particle is reflected (algorithmic response); if the particle has not yet returned into the domain, the collision detection and response procedures are repeated until the particle returns to the flow region. The collision detection and response algorithm brings the particles back into the domain since, in a real situation, fluid particles cannot escape to due to the presence of the boundaries (no-penetration condition).

This paper is organised as follows. Section 2 presents the methodology and implementation of the collision detection and response algorithm. Section 3 presents the mathematical modelling (with the aspects of analytical geometry applied in the algorithm). The validation of the algorithm is presented in Section 4. Sample applications of this computational tool within solid and fluid mechanics are described in Section 5. Finally, conclusions and directions for future work are presented in Section 6.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

2 IMPLEMENTATION METHODOLOGY

2.1 Particle Collisions Against a Single Plane

The input data of the algorithm are the equations of the planes that define the problem geometry, obtained from principles of analytical geometry; the initial and final positions of the centers of mass of all particles and the particles' velocities at each time step.

After obtaining the plane equations, the steps below are performed at each iteration.

1. *Calculation of the distance from the particle to the plane*. A plane has a positive and a negative region; the positive side is the direction in which the normal unit vector **n** points (Fig. 1). At the initial instant, the particle will always be in the positive region of the plane.

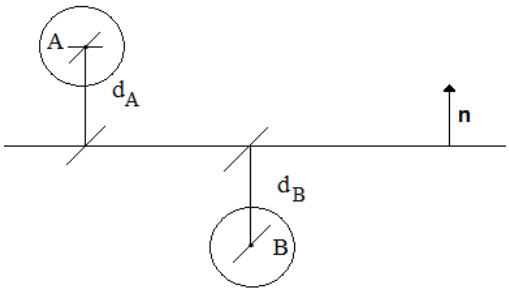

Figure 1. Positive and negative sides of the plane (where the centres of mass A and B are located, respectively).

2. *Collision detection.* There are three possible particle positions in relation to the plane in the case of free movement. They are: (A) the particle has not reached the spatial position of the plane; (B) the particle has reached the spatial position of the plane, but has not passed through it completely; and (C) it has passed through the plane entirely. Figure 2 shows these three possible situations. In this figure, d is the distance between the centre of mass of the particle and the plane, r is the radius of the particle, to is the initial time instant, C_o and C_1 are the positions of the centres of mass at the beginning and end of the numerical iteration (at time (to + Δt)) obtained by numerical integration for free particle motion without the existence of obstacles. Collision occurs in two situations: for $d > 0$ and $d < r$, or for $d < 0$. In (a), $d > 0$ and $d > r$, and thus a collision against the plane has not occurred. In (B), d is positive and smaller than the radius, and a collision is imminent. In (C), d is negative and the particle is on the negative side of the plane, meaning that a collision has occurred.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

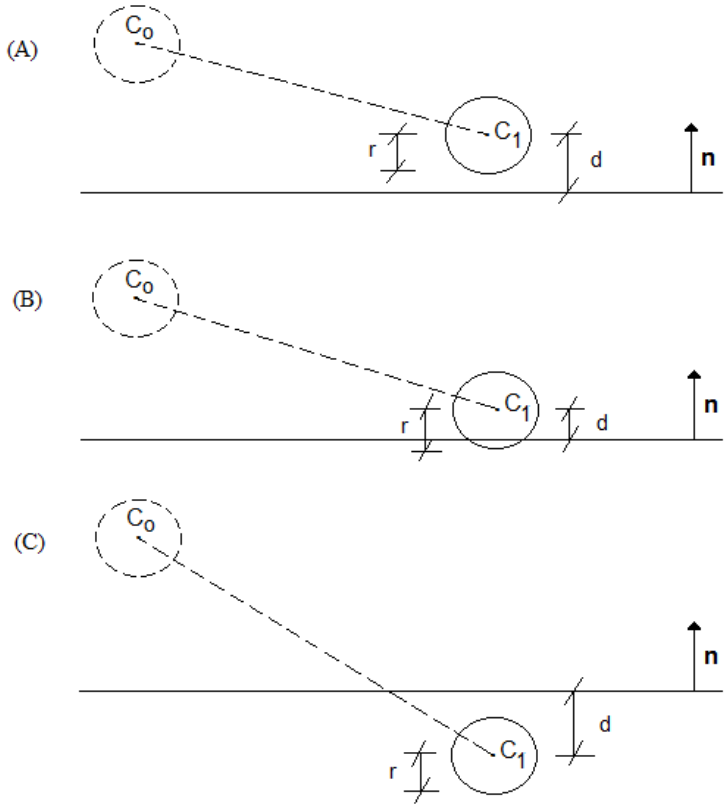

Figure 2. Collision test for a particle in relation to the lower plane (calculated from the analysis of the particle trajectory over a single time step).

3. *Collision response.* If a collision has occurred in the previous step, the particle must be relocated within the domain (enforcement of the non-penetration condition). This is accomplished by obtaining a vector **t**, which will shift the particle from C_1 to C_f (current position of the centre of mass of the particle, obtained after the collision response). The vector **t** has the same direction and sense as the vector **n**, and its modulus is given by Eq. (1):

$$|\mathbf{t}| = (1.0 + \mathrm{CR})(\mathrm{r} - \mathrm{d}) \tag{1}$$

where r > 0, d < 0, and CR is the restitution coefficient of kinetic energy ($1.0 \geq \mathrm{CR} \geq 0.0$).

The first point of the particle that collides is P_o. It is obtained by adding or subtracting the radius of the coordinate of C_o in the perpendicular direction to the collision plane. Figure 3 shows the points P_o and P_1, which represents the new position of the point at the instant ($t_o + \Delta t$), where Δt is the timestep. The distance a from P_1 to the collision plane is also shown.

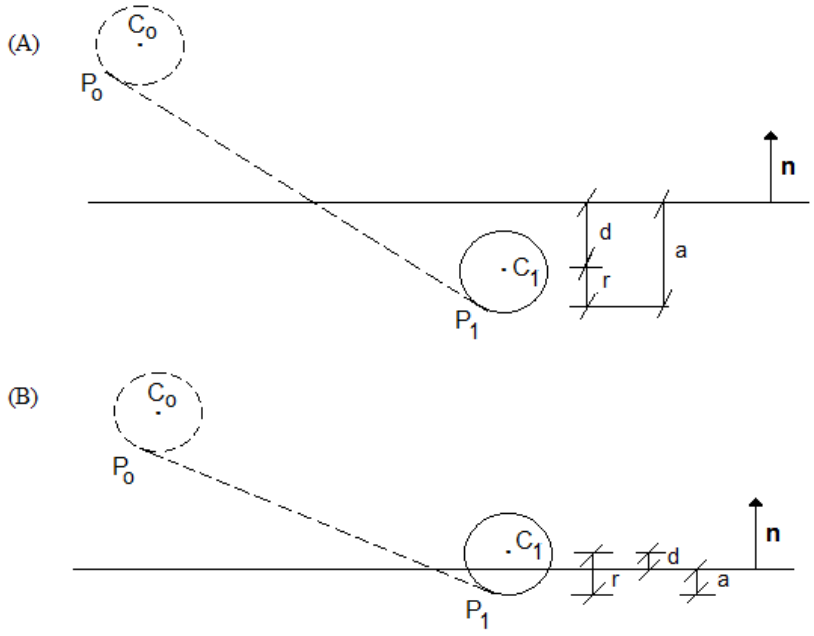

Figure 3. Distance between P_1 and the collision plane for free movement in two situations: (A) the particle traverses the plane completely; and (B) it partially traverses the plane.

Based on the vector **t**, the reflection of the particle is carried out; the position of the centre of mass at the moment of collision against the plane is C_I. The distance between C_1 and the axis (parallel to the plane and passing through C_I) is the same distance between this axis and C_f for a totally elastic collision, as shown in Fig. 4.

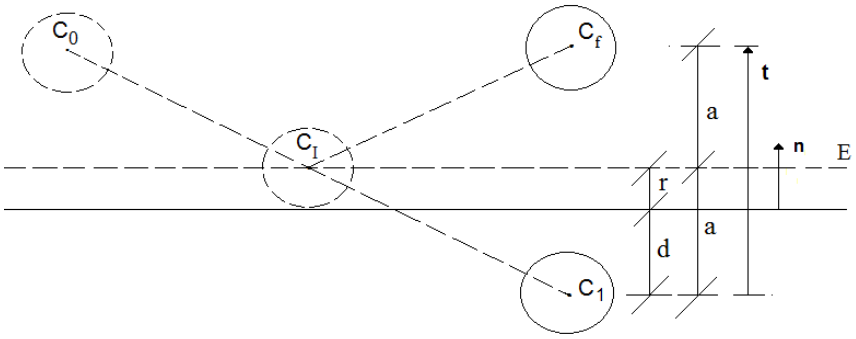

Figure 4. The reflection axis is given by E. The distance between C_1 and the axis E is equal to the distance between this axis and C_f for a totally elastic collision.

Finally, it is necessary to correct the magnitude of the velocity component perpendicular to the impact plane, according to Eq. (2):

$$(v_{col})_k = CR \times (v_p)_k \tag{2}$$

where:

$(v_p)_k$ and $(v_{col})_k$ are the magnitudes of the components of the particle velocities perpendicular to the collision plane, before and after the collision, respectively,

k is the perpendicular direction to the impact plane, $k \in (x,y,z)$.

After the collision, the sense of the component of the particle's velocity perpendicular to the wall is also corrected. The magnitude and sense of the velocity component parallel to the boundary remain the same.

The steps described in this subsection must be carried out for each particle in the domain.

2.2 Multiple Collision Planes

When there is more than one collision plane, the implementation is somewhat different. An additional consideration is necessary: the inside of the domain is the positive side of all the planes involved. The steps of the implementation are presented below:

1. *Determination of the plane that has the point of contact with the particle* (P_I) *closest to* P_o, *given its trajectory.* In this step, the possible collision planes are stored only if the distance between C_1 and the plane indicates a collision, as shown in Figs. 2 (B) and (C). The length of the line segment $P_I - P_o$ is calculated for the possible particle impact planes. The plane for which the length of the line segment is the smallest is the collision plane of the particle, as shown in Fig. 5.

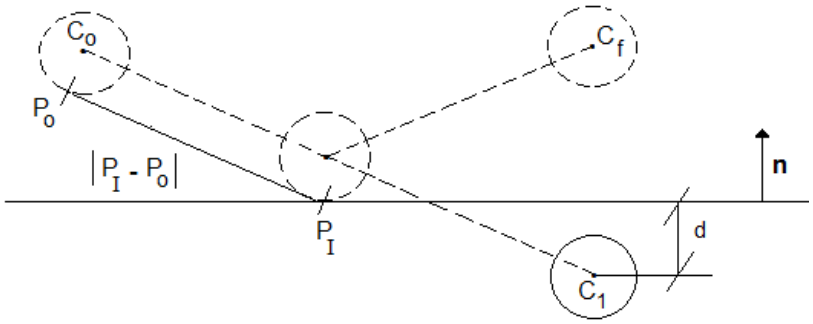

Figure 5. The smallest value of the length of the line segment P_I – Po defines the collision plane. The diagram shows this plane and the position of the centre of mass of the particle after the collision response.

2. *Collision response.* The algorithm performs the particle reflection based on the detected impact plane. After reflection, the particle trajectory becomes different from that at the beginning of the process. The initial point of the new trajectory is the centre of mass at the moment of contact with the plane of collision (C_o') and the end point is the point C_f obtained from the reflection (Fig.6).

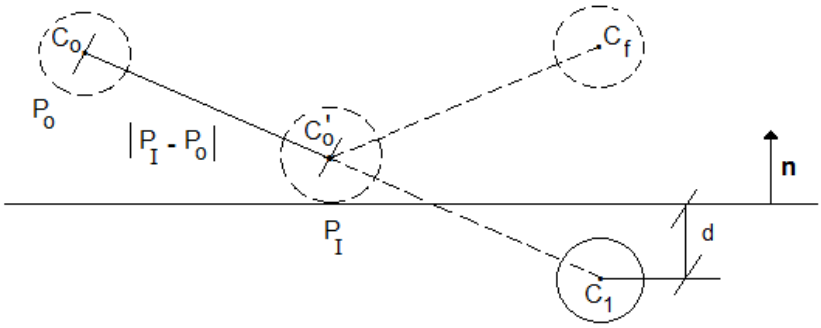

Figure 6. C_o' **is the new starting point of the particle trajectory after reflection, and** C_f **is the end point.**

If, after reflection, the particle is completely relocated within the domain (and is not in contact with any plane) the collision response procedure is complete.

3. *Once the points defining the new particle trajectory are known, steps 1 and 2 are repeated until no more collisions against a plane occur (the particle's centre of mass is repositioned within the domain in the current iteration).*

Figure 7 shows the successive collisions of the particle against planes A and B over one numerical iteration. Point C_1 is the final position that would be reached by the centre of mass of the particle if there were no walls delimiting the domain. In the first stage, the reflection of the particle is performed considering the plane B (the position of the centre of mass C_{1B} is obtained). The magnitude of the velocity component perpendicular to the boundary B is calculated and its sense is altered. Then, a second reflection is carried out (with a correction of the magnitude of the velocity component perpendicular to plane A) and the particle reaches the final position (centre of mass C_f).

All of the above steps should be done for each of the particles involved in the discretisation of the fluid.

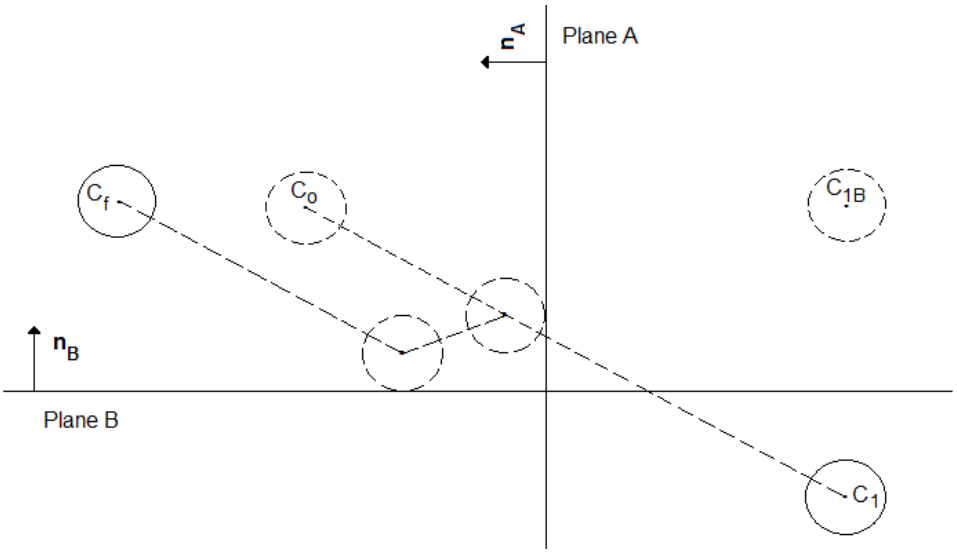

Figure 7. Collisions experienced by a particle in one time step.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering
P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

3 MATHEMATICAL MODELLING

The equations used in the implementation of the algorithm will be listed below, without further discussion, due to their simplicity and since they are basic principles of analytical geometry (Steinbruch & Winterle, 1987) .

In the notation used here, $X = (x_o, y_o, z_o)$ is a point in space determined by the position vector $\mathbf{r}_X = x_o\ \mathbf{i} + y_o\ \mathbf{j} + z_o\ \mathbf{k}$ (where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are the unit vectors in a 3D Euclidean space). In this way, a point can be seen as a vector.

The plane equation is given by Eq. (3):

$$\mathbf{n} \cdot (X - X_0) = 0 \tag{3}$$

where X is a point within the space and X_0 is a point belonging to the plane.

Equation (4) gives the distance between P_1 and a plane:

$$d = \mathbf{n} \cdot (P_1 - X_0) \tag{4}$$

where P_1 is the new spatial position of P_o at time instant $(t_o + \Delta t)$, for movement without obstacles.

The collision point P_I is given by Eq. (5):

$$P_I = P_0 + s\ \mathbf{t} \tag{5}$$

where:

$$\mathbf{t} = \frac{(C_1 - C_0)}{|C_1 - C_0|} \tag{6}$$

$$s = \frac{(X_0 - P_0) \cdot \mathbf{n}}{\mathbf{t} \cdot \mathbf{n}} \tag{7}$$

Since $\mathbf{t}$ is a unit vector, the distance between P_I and P_0 is equal to the value of the parameter s.

The coordinates of the position of the centre of mass must be corrected when the reflection is performed, as shown in Eq. (8):

$$(C_f)_k = (C_1)_k + (1.00 + CR)\,(r - d) \tag{8}$$

where: $(C_f)_k$ is the coordinate of the position of the centre of mass obtained after the collision response at the direction k,

$(C_1)_k$ is the coordinate of the position of the centre of mass at time instant $(t_o + \Delta t)$, in a motion without obstacles, at the direction k.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

4 CASE STUDIES

4.1 Static Physical Problems

In static problems, that is, for both solid and fluid mechanics, there is no motion of the rigid particles on a macroscopic scale. The algorithm was applied in two cases: rigid solid particles in equilibrium inside a box; and a Newtonian, incompressible and isothermal liquid (discretised using particles) at rest within a reservoir open to the atmosphere. Both cases used a 2D treatment.

Rigid Solid Spheres in Equilibrium

Figure 8 shows a transverse section of a box with sides of length unity, filled with rigid solid spheres of the same diameter.

Newton's first law states that if the forces acting on a body are balanced, the body is at rest (with velocity equal to zero), and this condition will only change by the action of an external force. According to Newton's second law of motion, if an unbalanced force acts on a body, it will experience acceleration (a change of velocity).

In the present case, the velocities and accelerations of the particles are zero at any instant in time. The solution of the momentum balance equation is very simple, and is in accordance with the laws of physics.

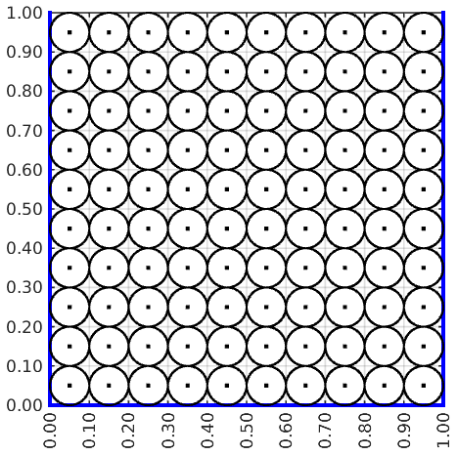

Figure 8. Particles in equilibrium inside a box.

After temporal integration, the input data (the initial and final positions of the centres of mass, coincidents, and zero velocities) are provided to the algorithm. At the collision detection stage, the algorithm detects contact between the particles and the lateral and lower planes (that is, the distance between the centre of mass of each particle and the plane is equal to the radius). In the response stage, the particle reflection is carried out using Eq. (8), and the position of the centre of mass is unchanged. The restitution coefficient of kinetic energy employed is zero, and the response to the particle velocities is also zero, maintaining the particles in contact with the static planes.

The results provided by the simulation using this algorithm are in agreement with the Newton's first and second laws.

Hydrostatic (Liquid at Rest Inside a Reservoir)

This hydrostatic problem consists of a tank that is at rest, open to the atmosphere, and filled with liquid. The dimensions of the transverse section are 1.0 m x 1.0 m. The origin of the frame of reference, with a positive vertical orientation pointing up, is fixed at the bottom of the reservoir. Figure 9 shows the distribution of the 50 Lagrangian discretisation particles (considered as rigid spheres) on each side of the tank.

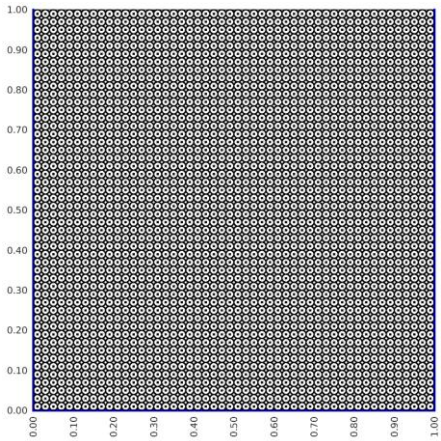

Figure 9. Initial particle setup.

In this fluid mechanics problem, the Smoothed Particle Hydrodynamics (SPH) method was used to solve the Navier-Stokes equation. The application of a macroscopic view and the continuum theory (where the molecular motion of the fluid is not considered) allowed the use of partial differential equations in the mathematical modelling. The physical laws of mass conservation and momentum balance (Newton's second law) were used in the solution of the problem.

The approximations of the physical properties used by the SPH method involve errors (Liu & Liu, 2010; Vaughan et al., 2008). In hydrostatic problems, the standard SPH formulation combined with a virtual particle boundary method gives rise to difficulties in guaranteeing the balance of forces and the unchanging positions of particles over time (Korzilius et al., 2014; Fourtakas et al., 2015). (Fraga Filho, 2017) guaranteed the balance of forces using the modified pressure concept, with ghost (virtual) particles located in an extended domain.

In this work, in order to obtain the input data for the computational algorithm, it is necessary to solve the momentum balance equation using the modified pressure concept (Batchelor, 2000). The 2,500 water particles inside the reservoir are at 20ºC and at sea level ($\rho = 1.00 \times 10^3$ kg/m^3, $\nu = 1.00 \times 10^{-6}$ m^2/s), where ρ and ν are the density and the kinematic viscosity of the water, respectively. The number of particles within the domain of

influence is twenty-one. Corrections to physical properties were not made in regions near the boundaries, where kernel truncation occurred. The time step used was 1.00×10^{-4} s.

Following the solution of the momentum balance equation and updating of the positions of the centres of mass and particle velocities, using numerical temporal integration at each numerical iteration, these remained unchanged (regardless of the kernel used) and the hydrostatic equilibrium was maintained.

When using the algorithm in the definition of the boundary conditions, it can be seen at the detection stage that contact occurs between particles and the lateral and lower planes. At the response stage, the reflections of the particles were carried out based on Eq. (8), and the positions of the centres of mass were unchanged. The restitution coefficient of kinetic energy employed was zero, and the responses to the velocities of the particles were also zero. There was the coincidence of the positions of the centres of mass at the final instant of the simulation (5.00 s/50,000 iterations) and their initial positions.

(Fraga Filho, 2017) presents a boundary treatment using virtual particles and modified pressure, which allows a consistent SPH solution to be obtained for the hydrostatic problem. In the present work, the use of this computational algorithm for the implementation of a physical reflective boundary, rather than virtual (ghost) particles, leads to a new solution which is also in agreement with the laws of motion and the continuum theory.

4.2 Dynamic Problem

Dam breaking studies are of great importance in the prevention of accidents, as these can cause serious environmental consequences and harm to residents in the vicinity. The fluid is considered to be incompressible, uniform and isothermal, and the Navier-Stokes equations, mass conservation and momentum balance must be solved.

The reservoir considered here has a length of 0.42 m, height of 0.44 m and depth of 0.228 m. Numerical simulations were performed for a rectangular area with a height of 0.228 m and width of 0.114 m, discretised using 2,556 fluid particles with a lateral distance between their centres of mass of 3.26 mm. In the simulation, the absolute pressure of the fluid was composed of the sum of a hydrostatic and a dynamic parcel (as predicted by Tait's state equation (Monaghan, 1994)). The artificial viscosity formulation was used with the coefficients α_π = 0.20 and β_π = 0.00 and a support radius varying with time, in order to avoid numerical instabilities and interpenetration between particles (Liu & Liu, 2003). The results were validated using experimental data (Cruchaga et al., 2007).

Figure 11 presents the simulated 2D computational domains and the initial distribution of the centres of mass of the particles.

The free surface particles were marked, and the constant pressure condition (zero) was applied and renewed at every numerical iteration. The time step used was 1.00×10^{-5} s. The simulations were run for a simulated time of 0.30 s (30,000 numerical iterations). Corrections were made to the physical properties (density and pressure gradient) of the particles located near the boundaries, using a correction factor based on the corrected smoothed particle method (Chen et al. 1997; Liu & Liu, 2010).

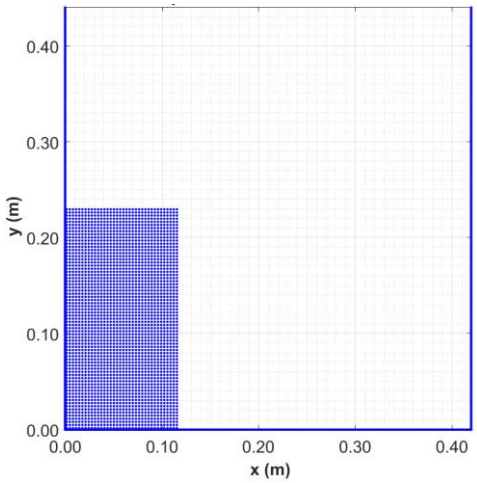

Figure 11. The simulated 2D computational domain and the initial distribution of the centres of mass of the particles for a breaking dam.

After carrying out the temporal integration using the Euler method, and obtaining the position of the centre of mass of each fluid particle (input data), the collision and response algorithm returned the particles colliding with the planes into the domain. The value of the restitution coefficient was 1.00. The wavefront position was monitored before its collision with the right-hand wall of the tank.

The greatest differences between the wavefronts (experimental and provided by the SPH/collision algorithm) occurred in the initial instants of the dam breaking. At 0.10 s, the difference was 6.90% (0.2032 m SPH/0.2170 m experimental). As time progressed, the difference decreased, and at time 0.20 s was 4.40% (0.3776 m SPH/ 0.3610 m experimental). Figure 12 shows a graphical illustration of the experimental and numerical results. The numerical results showed good agreement with the experimental results within the simulation period.

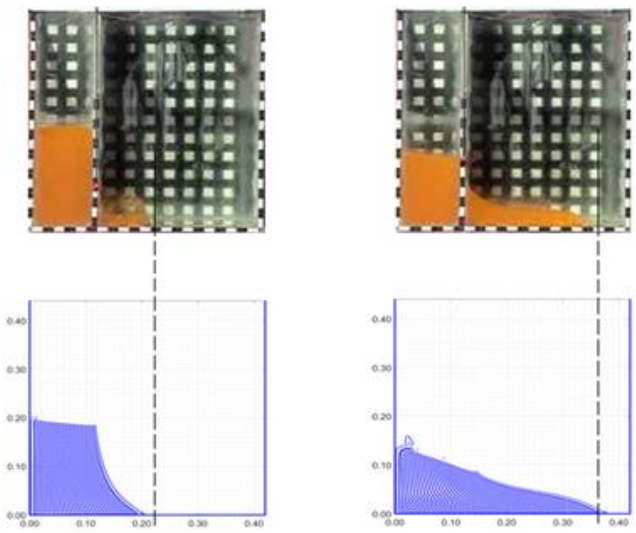

Figure 12. Experimental and numerical results for a breaking dam.

5 CONCLUSIONS

The algorithm presented in this paper was implemented, validated and then applied within solid and fluid mechanics problems. In a one-particle model, as in the standard Smoothed Particle Hydrodynamics (SPH) method, the collisions between the fluid particles in the flow are not considered. In recent years, (Korzilius et al., 2014) have proposed an inter-particle model for the SPH method and a boundary treatment using ghost particles, which have achieved good results. A solution for the hydrostatic problem based on SPH, using ghost or virtual particles was proposed by (Fraga Filho, 2017). The results achieved in this work show that physical reflective boundary conditions lead to excellent results in the static case and are in agreement with experimental data for the hydrodynamic problem.

The computational algorithm presented here can be generalised to various situations, such as planes forming non-perpendicular angles, through the simple choice of normal unit vectors to define the fixed planes in the contours, demonstrating its potential for application in other problems of Computational Fluid Mechanics (as well as Solid Mechanics).

REFERENCES

House, D. H., & Keyser, J. C., 2017. Foundations of Physically Based Modeling & Animation, CRC Press, Taylor & Francis Group, Boca Raton, Florida.

Kelager, M., 2006. Lagrangian Fluid Dynamics using Smoothed Particle Hydrodynamics. Department of Computer Science, University of Copenhagen, Denmark, available at http://glowinggoo.com/sph/bin/kelager.06.pdf, accessed August 3, 2017.

Steinbruch, A., & Winterle, P., 1987. Geometria Analítica, Editora Pearson Education, 2nd edition.

Liu, M. B., & Liu, G. R., 2010. Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments, *Arch. Comput. Methods Eng* **17**, pp. 25-76 (2010).

Vaughan, G. L., Healy, T. R., Bryan, K. R., Sneyd, A. D., & Gorman, R. M., 2008. Completeness, conservation and error in SPH for fluids, *International Journal for Numerical Methods in Fluids* **56**, pp. 37–62.

Korzilius, S. P., Kruisbrink, A. C. H., Schilders, W. H. A., Anthonissen, M. J. H., & Yue, T., 2014. Momentum conserving methods that reduce particle clustering in SPH, *CASA-Report 2014-2015*, (Technische Universiteit Eindhoven, Eindhoven, 2014), available at https://pure.tue.nl/ws/files/3858217/376670351851652.pdf, accessed July 30, 2017.

Fourtakas G., Vacondio R., & Rogers B. D., 2015. 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, *International Journal for Numerical Methods in Fluids*, **78** (8), pp. 475-501 doi: 10.1002/fld.4026.

Fraga Filho, C. A. D., 2017. Development of a computational instrument using a Lagrangian particle method for physics teaching in the areas of fluid dynamics and transport phenomena, *Rev. Bras. Ensino Fís*. (online) **39** (4), e4401 (2017), available at http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172017000400501&lng=en&nrm=iso, accessed July 30, 2017.

Batchelor, G.K., 2000. An Introduction to Fluid Dynamics. Cambridge University Press, 3rd edition.

J. J. Monaghan, 1994. Simulating free surface flows with SPH, *Journal of Computational Physics* **110** (2), pp. 399-406.

Liu, G. R., & Liu, M. B., 2003. Smoothed Particle Hydrodynamics: a Meshfree Particle Method, World Scientific, Singapore.

Cruchaga, M. A., Celentano, D. J., & Tezduyar, T.E., 2007. Collapse of a liquid column: numerical simulation and experimental validation, Computational Mechanics **39**, pp. 453-476.

Chen, J. K., Beraun, J. E., & Carney, T. C., 1997. A corrective smoothed particle method for boundary value problems in heat conduction, *Int. J. Num. Meth. Eng.* **46**, pp. 231–252.