

# Numerical Analysis of a Dam-Break Problem Using Smoothed Particle Hydrodynamics with Fourth Order Polynomial Smoothing Functions

Saleen Bhattarai <sup>a</sup>, Laxman Poudel <sup>b</sup>, Hari Bahadur Dura <sup>c</sup>

<sup>a, b, c</sup> Department of Mechanical Engineering, Pulchowk Campus

Corresponding Email: <sup>a</sup> saleenbhattarai@gmail.com, <sup>b</sup> laxman@ioe.edu.np, <sup>c</sup> duraharis@gmail.com

## Abstract

Smoothed Particle Hydrodynamics (SPH) has been extensively used to model and simulate free surface flows. Here, SPH is used to discretize and solve the Euler equations for a simple dam-break case employing fourth order polynomial smoothing functions for approximating the first derivatives. Repulsive particles are used to prevent nonphysical boundary penetration and virtual particles to ensure completion of the summation. Significant improvements in boundary momentum contributions (6.33 MN to 1.27 MN) and impulses (633 Ns to 63 Ns) are observed as a result reducing the time-step by half and introducing virtual particles. It is also found that a lower value of Lennard-Jones parameter ( $D = 24.53 \text{ m}^2/\text{s}^2$  to  $4.91 \text{ m}^2/\text{s}^2$ ) reduces the nonphysical splashing behavior (0.9 m to 0.48 m). Finally, it is concluded that the choice of smoothing function (for ensuring similar solver performance) is dependent on the temporal scale of the problem.

## Keywords

Meshless, Particle, SPH, Dam Break, Boundary

## 1. Introduction

The numerical description of any flow is done through the solution of relevant governing partial differential equations (PDE) by either mesh-based or meshless methods. The former are always bound in their accuracy by the choice of underlying mesh [1] and are limited in only qualitative accuracy when dealing with multi-phase flows [2]. Dependence of the solution in mesh is eliminated in entirety by using meshless methods through robust and unique particle based description of constituent phases [1].

Meshless method in the form of SPH was first utilized to model boundaryless galactic phenomenon [3]. Over the years, several works have realized the potential of SPH in solving fluid flow problems [1]. The method itself has underwent several improvements to increase its accuracy [4] and modifications to accommodate the definition of boundaries [5, 2].

A robust nature of the method itself allows for an easy definition of multi-phase flows involving water and sediment. Owing to the present context of sediment-laden Nepalese rivers [6] being used for hydropower, the potential for studying effects of sediment erosion

in turbines using SPH is immense.

Here, we apply the techniques of SPH and particle method to discretize and numerically solve Euler equations. The problem is taken to be a dam-break case as it is simple, popular and easily comparable [2].

Recent studies in this area have revealed the effectiveness of filtering techniques in reducing unwanted density oscillations which may adversely affect the pressure fields and surface profiles [7]. The preference in using cylindrical particles is prevalent when using shallow water equations which, however, comes with a marginal error in local 3D flow reproduction immediately near the dam-break site [8]. Comparison of non-dimensionalized solver results with analytical and experimental results have shown the effectiveness of SPH in this area [9].

## 2. Methodology

### 2.1 Mathematical Modeling

When dealing with any flow, the Navier-Stokes equations need to be solved in order to get the solution of that particular problem. For particle methods, the

Lagrangian form of these equations are required as written below.

$$\begin{bmatrix} \frac{D\rho}{Dt} \\ \frac{Dv^\alpha}{Dt} \\ \frac{De}{Dt} \end{bmatrix} = \begin{bmatrix} \rho \frac{\partial v^\beta}{\partial x^\beta} \\ \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} + \frac{F^\alpha}{\rho} \\ \frac{\sigma^{\alpha\beta}}{\rho} \frac{\partial v^\alpha}{\partial x^\beta} + Q \end{bmatrix} \quad (1)$$

Here,  $\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta}$  represents the total stress tensor with  $\delta$  as the Dirac-Delta function;  $\tau^{\alpha\beta} = \mu\varepsilon^{\alpha\beta}$  represents the viscous shear stress with  $\mu$  as the dynamic viscosity and  $\varepsilon$  as the viscous strain rate. If we consider the flow to be inviscid, then we can ignore all terms related to  $\tau^{\alpha\beta}$  and simplify Equation (1) into the Euler equations. Furthermore, if we consider the flow to be non-generative, then we can also ignore  $Q$  to further simplify it.

### 2.1.1 Integral Approximation

The main principle of SPH lies in the approximation of a field variable  $f$  at a location  $x$  using the corresponding values of  $f$  at neighboring locations  $x'$  by 'weakening' or 'smoothing' the influence of those neighboring locations through the implementation of a compactly defined smoothing kernel function  $W$ .

$$\begin{bmatrix} f(x) \\ f'(x) \end{bmatrix} = \begin{bmatrix} \int_{\Omega} f(x')W(x-x',h)dx' \\ I - \int_{\Omega} f(x')W'(x-x',h)dx' \end{bmatrix} \quad (2)$$

Here,

$$I = \int_S f(x')W(x-x',h).\hat{n}dx' \quad (3)$$

simplifies into the compact support property (boundary condition) of the smoothing kernel function which must be zero for  $x-x' \geq h$ . Also, as seen in Equation (2), only  $W$  needs to be differentiated to  $W'$  in order to approximate  $f'$ .

### 2.1.2 Particle Approximation

The integral from in Equation (2) can be converted into a discrete or a particle form, as shown in Equation (4), by using a summation, whereby the value of  $f$  for particle  $i$  is influenced by the corresponding values of  $f$  for particles  $j$  which lie inside the domain of influence  $h$  of the former. The degree of influence is proportional to  $W'(x-x',h)$ . In order to make the equations consistent in dimensions,

volume  $V_j$  is multiplied in the R.H.S.

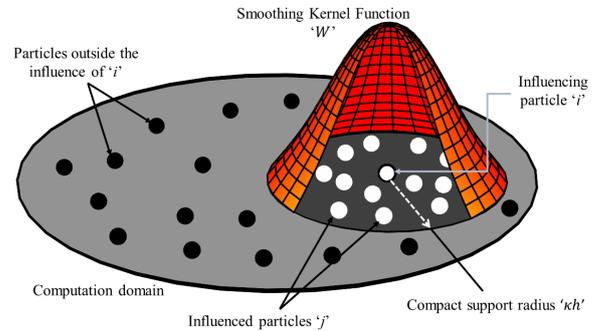
$$\begin{bmatrix} \langle f(x_i) \rangle \\ \langle f'(x_i) \rangle \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N f(x_j)W(x_i-x_j,h)V_j \\ I - \sum_{j=1}^N f(x_j)W'(x_i-x_j,h)V_j \end{bmatrix} \quad (4)$$

Implementing the concepts of Equation (4) to discretize the Euler form of Equation (1), we obtain the following.

$$\begin{bmatrix} \frac{D\rho_i}{Dt} \\ \frac{Dv_i^\alpha}{Dt} \\ \frac{De_i}{Dt} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N m_j v_{ij}^\beta \cdot \frac{\partial W_{ij}}{\partial x_i^\beta} \\ \sum_{j=1}^N m_j A_{ij}^{\alpha\beta} \frac{\partial W_{ij}}{\partial x_i^\beta} + \frac{F^\alpha}{\rho_i} \\ \frac{1}{2} \sum_{j=1}^N m_j B_{ij}^{\alpha\beta} \frac{\partial W_{ij}}{\partial x_i^\beta} \end{bmatrix} \quad (5)$$

Here,  $v_{ij}^\beta = v_i^\beta - v_j^\beta$  represents the symmetrized form of velocity. The expressions of  $A_{ij}^{\alpha\beta}$  and  $B_{ij}^{\alpha\beta}$  (both equal to one another under the current assumption scheme discussed earlier) in Equation (5) have been expanded below.

$$A_{ij}^{\alpha\beta} = B_{ij}^{\alpha\beta} = \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \quad (6)$$



**Figure 1:** Particle approximation in SPH

## 2.2 Numerical Methods

Fourth order polynomials satisfying the conditions of normalization, compactness, Dirac-delta approach, positivity, decay, continuity and symmetry are naturally fit to be used as smoothing kernel functions [10]. The reproduction of  $f$  and  $f'$  will naturally be fourth order accurate as these polynomials satisfy the moment  $M$  criteria for  $M_0, M_1, M_2, M_3$  and  $M_4$ . [10] Likewise, satisfaction of the same conditions ensures fourth order consistency. However, since particle approximation (instead of integral) has been considered (Equation (4) and (5)),

the consistency is largely determined by the choice of initial particle distribution.

Based on the above facts, the following non-piecewise functions have been selected. Note that it is easier to write  $W$  in terms of non-dimensional radial distance between an interacting pair  $R = \frac{r}{h}$ ; where we define  $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  to be the absolute radial distance.

### 2.2.1 Quartic Function [3]

It has a well defined compact support in the range  $0 \leq R \leq 1$  for both first and second derivatives.

$$W(R, h) = \alpha_d (1 + 3R)(1 - R)^2 \quad (7)$$

Where  $\alpha_d$  is the surface integral parameter whose value depends on dimension of the problem as follows.

$$\alpha_d = \begin{cases} \frac{5}{4h} & ; 1D \\ \frac{5}{\pi h^2} & ; 2D \\ \frac{105}{16\pi h^3} & ; 3D \end{cases} \quad (8)$$

### 2.2.2 New Quartic Function [10]

It has a well defined compact support for the first derivative in the range  $0 \leq R \leq 2$  but not for the second derivative. However, it is more accurate due to a higher center weighting and more stable due to a smoother second derivative.

$$W(R, h) = \alpha_d \left( \frac{2}{3} - \frac{9R^2}{8} + \frac{19R^3}{24} - \frac{5R^4}{32} \right) \quad (9)$$

$$\alpha_d = \begin{cases} \frac{1}{h} & ; 1D \\ \frac{15}{7\pi h^2} & ; 2D \\ \frac{315}{208\pi h^3} & ; 3D \end{cases} \quad (10)$$

### 2.2.3 Gradients

For two dimensional case, the gradients of both smoothing functions have been calculated in Equations (11) and (12). These can be substituted into Equation (5) to obtain the fully discrete and program-ready form of the Euler equations.

$$\frac{\partial W_{ij}(R)}{\partial x_i^\beta} = \frac{-60}{\pi h_{ij}^4} x_{ij}^\beta (1 - R_{ij}) \quad (11)$$

$$\frac{\partial W_{ij}(R)}{\partial x_i^\beta} = \frac{-15}{28\pi h_{ij}^4} x_{ij}^\beta \left( 9 - \frac{19R_{ij} - 5R_{ij}^2}{2} \right) \quad (12)$$

Here,  $x_{ij}^\beta = x_i^\beta - x_j^\beta$  represents the relative position of  $i$  with respect to  $j$ .

## 2.3 Boundary Conditions

We cannot clearly define boundaries in particle methods, unlike FEM or FVM where boundary represents a unique physical barrier for a field variable. Researchers have determined many workarounds to accurately fix this limitation [2, 11, 12]. A combination of two such methods, also shown in Figure 2, where unique particles are introduced in location of the physical boundary has been discussed here.

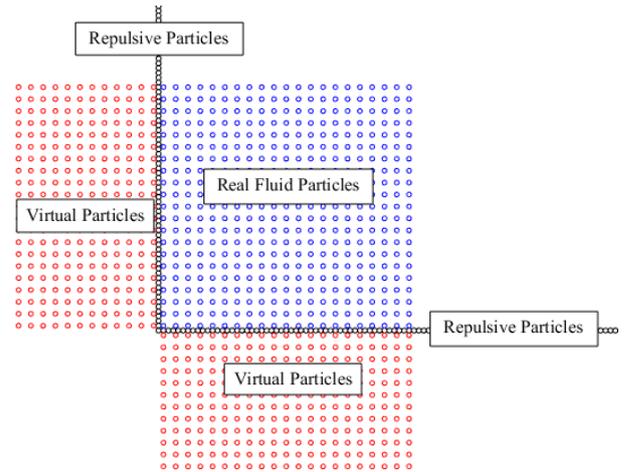


Figure 2: Boundary definition

### 2.3.1 Repulsive Particles

These particles are located permanently on the boundary and do not contribute to the summations in Equation (5). Any real fluid particle must never cross over these. To ensure this, these particles must repel any approaching real fluid particle according to a Lennard-Jones force [5] defined in the following way.

$$F_{ij}^\beta = D \left\{ \left( \frac{r_0}{r_{ij}} \right)^{n_1} - \left( \frac{r_0}{r_{ij}} \right)^{n_2} \right\} \frac{x_{ij}^\beta}{r_{ij}^2} \quad (13)$$

Here,  $D$  is a problem dependent parameter proportional to square of the velocity scale,  $n_1, n_2 = 4, 2$  and  $r_0$  is an optimizable cutoff distance such that  $F_{ij} = 0$  when  $r_{ij} \geq r_0$ .

### 2.3.2 Virtual Particles

These particles are placed immediately outside the boundary and are updated for every particle iteration. They are spatial and kinematic mirrors, and hydrodynamic homologues of real fluid particles located sufficiently close to the boundaries [2]. Their function is to simply ensure completeness of the summations in Equation (5) of particles that are close to the boundary.

### 2.4 Program Algorithm

MATLAB has been used for programming the SPH solver and simulating various cases. The program follows an iterative calculation process for determining required derivatives at each time-step after which they are time-marched using a constant time-interval simple integration scheme. The general flowchart of the code is outlined in Figure 3.

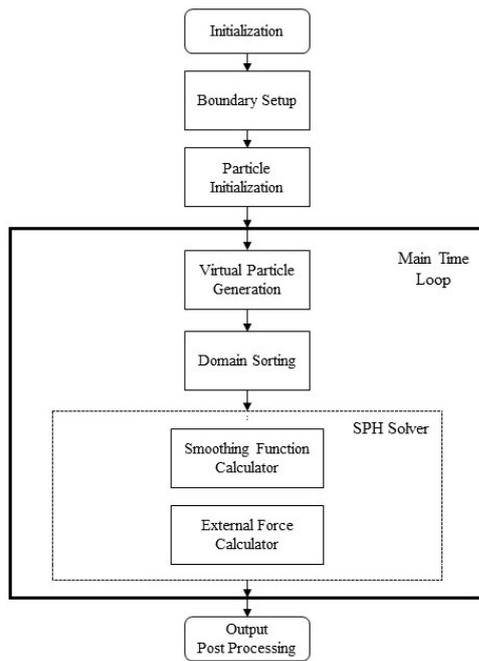


Figure 3: Program flowchart

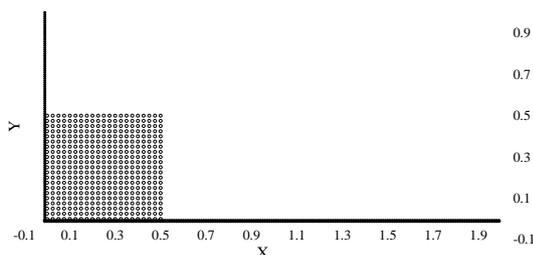


Figure 4: Initial setup of the particles

### 2.5 Case Setup

The dam-break problem contains a column of water which is allowed to free-fall. The domain has got two free-slip wall boundaries called ‘Wall’ (1 m) and ‘Floor’ (2 m) each with 100 and 200 uniformly spaced repulsive particles. A square water column (0.25 m<sup>2</sup>) is present with 441 uniformly distributed water particles ( $\Delta x = 0.025$  m) which are initially assumed to be in hydro-static equilibrium. The initial particle setup is shown in Figure 4. Three cases have been setup for simulation which are Dam Break 1 (DB1), Dam Break 2 (DB2) and Dam Break with Virtual Particles (DBVP). The hydrodynamic and kernel properties selected for the three cases have been summarized in Table 1. It is noted that  $D \propto gH$  [5]. Also,  $r_0 = 0.48\Delta x$  [13].

Table 1: Different simulated cases

Case	DB1	DB2	DBVP
$\Delta t$	0.001 s	0.001 s	0.0005 s
$W$	Quartic	New Quartic	Quartic
$h_s$	0.3 m	0.6 m	0.3 m
$n_1, n_2$	4, 2	4, 2	4, 2
$r_0$	$0.48\Delta x$	$0.48\Delta x$	$0.48\Delta x$
$D$	$5gH$	$2gH$	$gH$
VP	×	×	✓
$ x - x^* _{\rho,u,v,e}$	0.001	0.001	0.0001

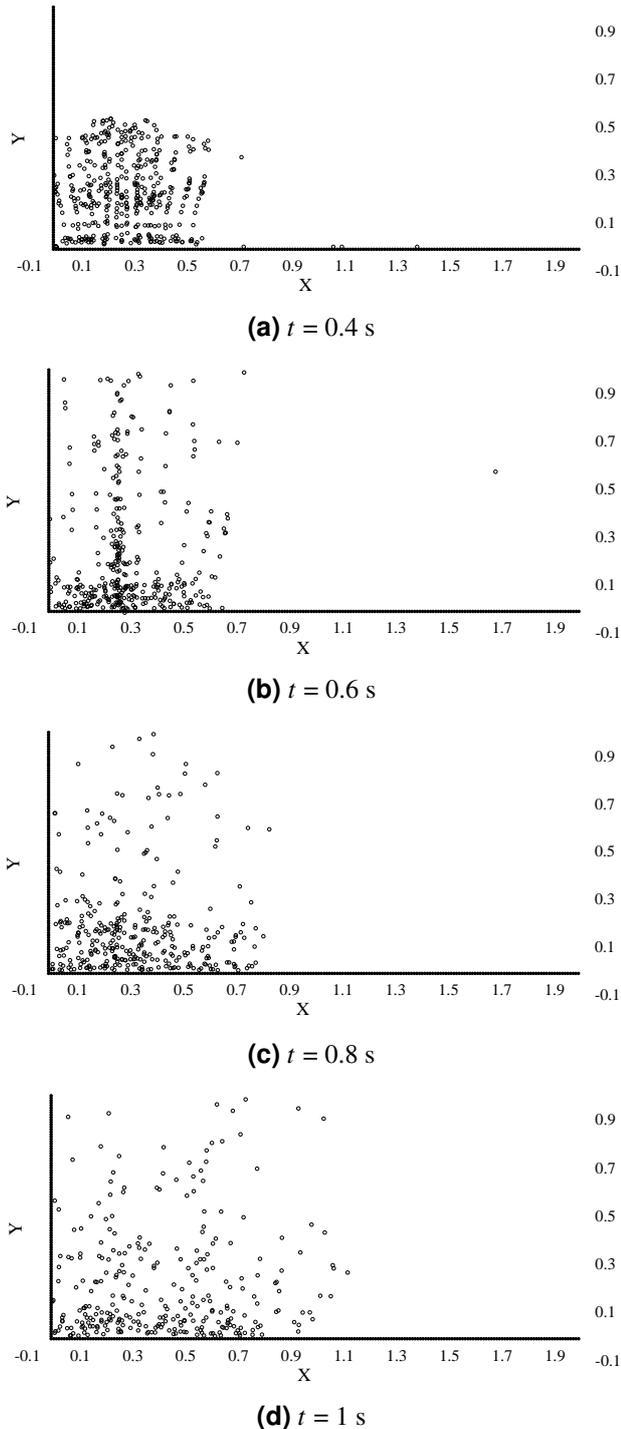
## 3. Results and Discussion

### 3.1 DB1

The results of this case are shown in Figure 5. The column of water particles is observed to fall towards the floor as expected. Some of them approaching the boundary get propelled upwards by repulsive particles via Lennard-Jones force, as a result of which, no particle is observed to cross the boundary. A localized splashing behavior is seen at  $t = 0.6$  s where particles are observed to nonphysically splash over their initial maximum height of 0.5 m and reach up-to 0.9 m as seen in Figure 5b. The front of water column is observed to traverse a distance of 0.6 m throughout the simulation duration of 1 s.

The reasons for nonphysical splashing are twofold. Firstly, the boundary momentum contribution of repulsive particles is seen to be an average of 6.33 MN for fluid particles closer than  $r_0 = 0.48\Delta x = 0.012$  m. Secondly, the average impulse for those same particles is determined to be 633 Ns. The former is caused due to a high value of  $D = 5gH =$

24.525 m<sup>2</sup>/s<sup>2</sup>. The latter is caused due to a coarser time-step ( $\Delta t = 0.001$  s) which reduces the number of intermediary steps available to any water particle approaching boundary for adjusting and reacting to such large impulses.



**Figure 5:** Solution of DB1 at different time-steps

### 3.2 DB2

The results of this case are shown in Figure 6. Similar to the previous case, no particles have directly penetrated the boundary. However, unlike DB1 where a middle column of particles exhibit splashing behavior, an entire particle front is observed to get splashed upwards at  $t = 0.4$  s and  $0.6$  s as seen in Figures 6a and 6b. The maximum height of the splash is again larger than expected (0.9 m at  $t = 0.6$  s). The majority of particles are seen to exhibit a flow like behavior whereby the water column recedes downwards and appears to spread over the floor by 0.6 m similar to DB1. Average height of the column, however, reduces to 0.26 m as seen in Figure 6d.

Compared to DB1, the flow behavior is seen to be more physical. Firstly, the localized splashing is eliminated as a result of using New Quartic smoothing function whose superior domain of influence discussed in Section 2.2.2 ends up affecting more particles. Moreover, the improvement in flow behavior is attributed to a reduction in average boundary momentum contribution to just 0.89 MN (average impulse of 90 Ns) which is due to a lower value of  $D = 2gH = 9.81$  m<sup>2</sup>/s<sup>2</sup> and a larger value of the smoothing length ( $h_s = 0.6$  m).

### 3.3 DBVP

The computational cost is observed to highly increase with the introduction of virtual particles. Thus, reversion of smoothing function from New Quartic back to Quartic is paramount for maintaining a similar solver performance.

The results of this case are shown in Figure 7. The boundary induced splashing phenomenon at  $t = 0.4$  s and  $0.6$  s, as seen in Figures 7a and 7b, is significantly reduced compared to previous cases. Moreover, the splashing is observed to be uniform and physically consistent as no particle crosses a maximum height of 0.5 m. Compared to DB2, an even better flow behavior is seen as the front traverses 0.6 m, recedes to an average height of 0.2 m in  $t = 1$  s and 2 m along with 0.15 m of the same in  $t = 2$  s shown in Figures 7d and 8 respectively.

The massive improvements in flow behavior is largely contributed by the introduction of virtual particles. Boundary momentum contribution is found to be an average of 1.27 MN which is larger than DB2 in spite of using  $D = gH = 4.91$  m<sup>2</sup>/s<sup>2</sup> because of reverting back to the less superior Quartic function. However,

despite being greater, an even finer time-step ( $\Delta t = 0.0005$  s) is found to reduce the average impulse to just 63.3 Ns (ten times less than DB1). As a result, the particles are observed to exhibit smoother flow behavior due to them getting more intermediary steps to adjust and react to the boundary impulse.

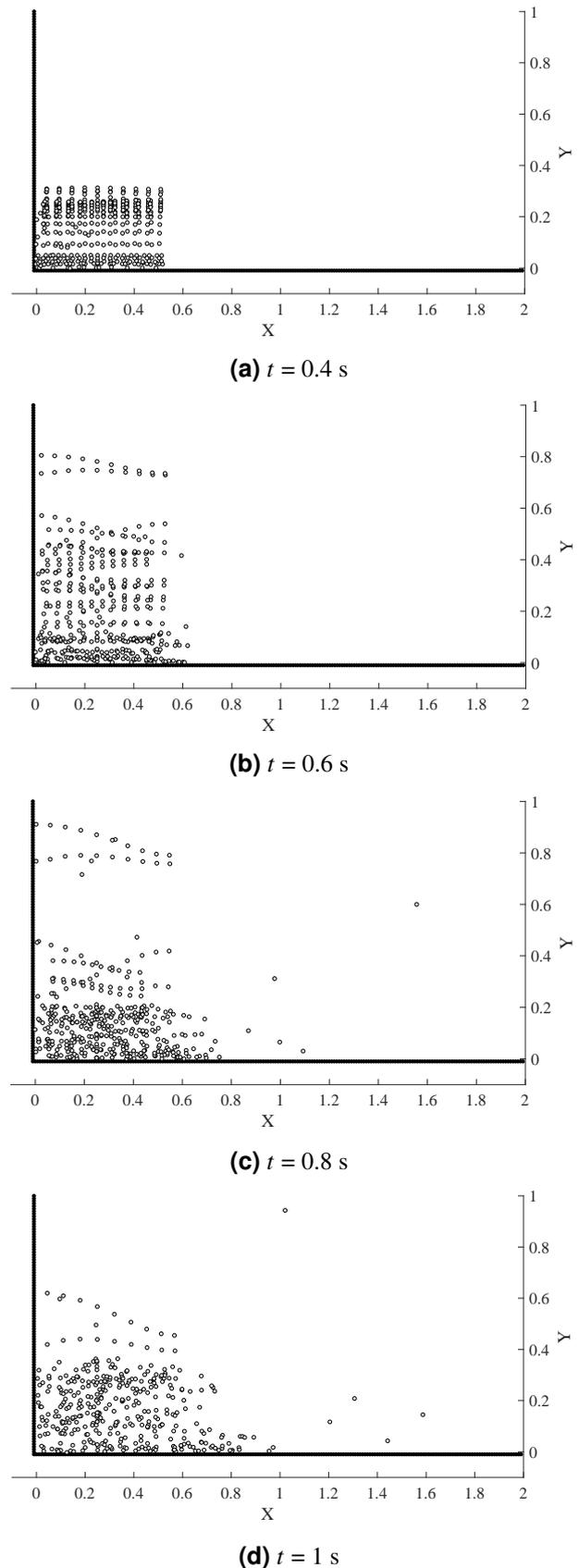
### 3.4 Validation and Comparison of Results

For validation, comparing the results of current solver are compared with experimental data [5] and previously obtained numerical results [10], it is better to non-dimensionalize the observations of front and level of the water column and the elapsed time using initial height ( $H_0$ ) and characteristic time ( $T_0 = \sqrt{H_0/g}$ ) respectively. The comparisons have been presented in Figure 9. While the trend-line of front in Figure 9 shows a similar shape to both experimental and numerical results, ignoring viscous effects in the simulation has caused the results from current solver to experience a temporal shift due to the particles not dissipating their energy quick enough. The results for level in Figure 9 are observed to be more satisfactory and don't experience the temporal shift. However, oscillations are observed initially due to the particles interacting conservatively with the boundary due to a lack of viscosity.

Likewise, the velocity profiles in y-axis obtained from the current solver at two locations along the flow when  $t/T_0 = 10$  are also compared in Figure 10 with analytical (Ritter), experimental (Exp) and previously obtained numerical (Num) results [9]. All positions have been non-dimensionalized using  $H_0$  and velocities using characteristic wave velocity ( $U_0 = \sqrt{gH_0}$ ). It is seen from both cases of Figure 10 that the obtained numerical results from the current solver show good agreement with analytical, experimental and existing numerical results for  $y/H_0 \geq 0.2$  but not for lower regions. The main reason for this is that the existing numerical results consider viscous interactions as well. This leads to strong velocity gradients near the bottom wall possibly indicating formation of a boundary layer like region. This is not possible in the results from current solver as viscosity has not been considered.

## 4. Conclusion

This study has illustrated the process of discretizing the Euler equations using SPH in order to numerically solve a dam-break case. Fourth order smoothing



**Figure 6:** Solution of DB2 at different time-steps

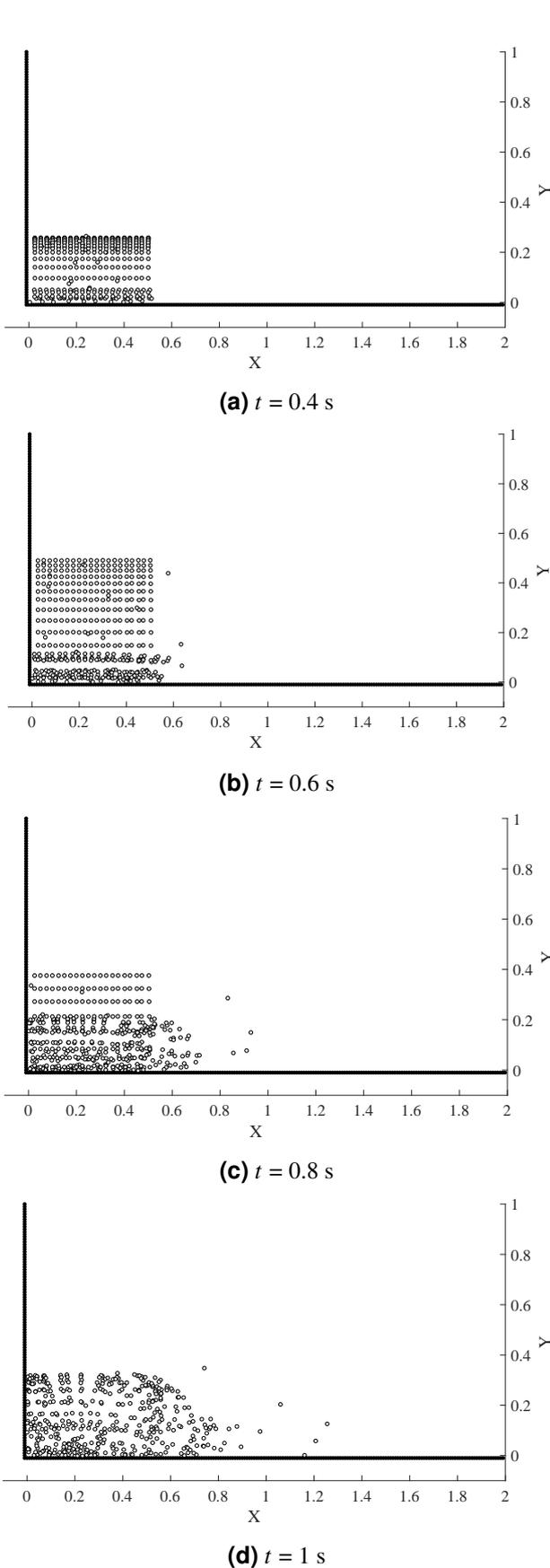


Figure 7: Solution of DBVP at different time-steps

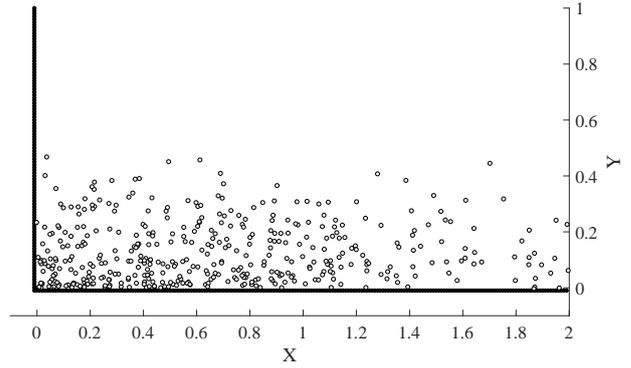


Figure 8: Solution of DBVP at  $t = 2$  s

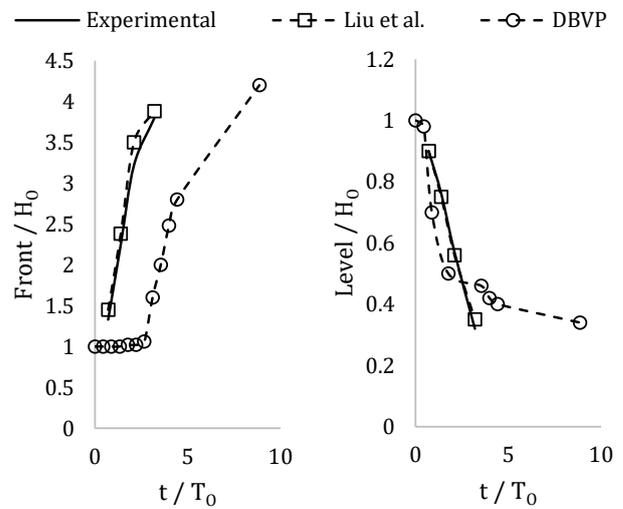


Figure 9: Comparison of normalized front and level

functions are used for the required purpose. As a consequent, the developed MATLAB code is capable of solving weakly compressible two dimensional inviscid flows.

It has been concluded from obtained results that the value of  $D$  should be in the order of  $gH$  when dealing with a free surface flow in order to maintain the boundary momentum contributions at acceptable levels to prevent nonphysical splashing. Likewise, it also has been concluded that time-step of the simulation should be reduced in order to decrease the repulsive impulse and allow more number of intermediary steps for particles to smoothly adjust themselves near the boundary. Moreover, it has been demonstrated that the introduction of virtual particles near the boundaries improves flow behavior through a more pronounced front spreading. Furthermore, New Quartic function is determined to be more accurate than Quartic function. However, due to differences in their domain of influence, their choice to a particular problem depends on a reciprocal trade-off with

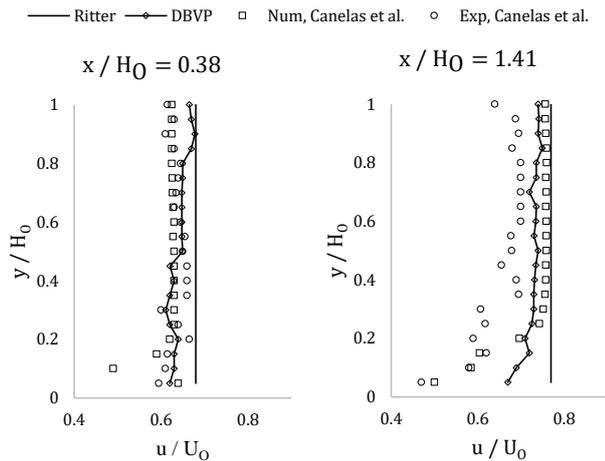


Figure 10: Comparison of velocity profiles in y-axis

time-step for maintaining a similar solver performance. Finally, in spite of the temporal shift and oscillation inaccuracy of the present solver concluded to be a result of an inviscid assumption, the velocity profiles obtained from the solver have been observed to show reasonable agreement with existing results.

## 5. Future Works

A numerical analysis naturally has inherent limitations governed by the necessary assumptions. Based on these, several prospects and recommendations for future works still remain. Firstly, the flow can be treated as viscous whereby all the viscous terms in Equation (1) may also be discretized. Secondly, the developed MATLAB code may easily be extended into three dimensions. Thirdly, simultaneous solution of Equation 5 for another phase can easily convert this solver into multi-phase which may be used to analyze behavior of water-sediment flows. Finally, porting the code to a parallel process for faster simulation times along with unambiguous definition of virtual particles at discontinuous boundary locations is still a challenge.

## Acknowledgments

The authors would like to acknowledge the support of Incubation, Innovation and Entrepreneurship Center (IIEC) for providing the necessary platform and computational resources for carrying out the work.

## References

- [1] Antonio Huerta, Ted Belytschko, Timon Rabczuk, and Sonia Fernández-Méndez. *Meshfree methods*, volume 1, chapter 10, pages 279–309. John Wiley and Sons, Ltd, 2004.
- [2] Andrea Colagrossi and Maurizio Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics*, 191(2):448–475, 2003.
- [3] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82(12):1013–1024, 1977.
- [4] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.
- [5] J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2):399–406, 1994.
- [6] Laxman Poudel, Bhola Thapa, and Bhim Prasad Shrestha. Sand shape analysis with erosion on hydraulic turbine material: A case study of Roshi khola of Nepal. *Kathmandu University Journal of Science, Engineering and Technology*, 9(2):34–46, 2013.
- [7] Nabeel Abed. evaluation of some treatments in the dam-breaking flow using the smoothed particles hydrodynamics (sph) approach. *International Journal of Petrochemical Science and Engineering*, 3(3):102–107, 2018.
- [8] Hong-Ming Kao and Tsang-Jung Chang. Numerical modeling of dambreak-induced flood and inundation using smoothed particle hydrodynamics. *Journal of Hydrology*, 448-449:232–244, 2012.
- [9] Ricardo Canelas, R. Alexio, and Rui M L Ferreira. Sph-based numerical simulation of the velocity field in a dam-break flow. In *MEFTE 2013*, 04 2012.
- [10] M. B. Liu, G. R. Liu, and K. Y. Lam. Coupling meshfree particle method with molecular dynamics novel approach for multiscale simulations. In G. R. Liu, editor, *Advances in Meshfree and X-FEM Methods*, volume 2, pages 211–216. World Scientific, 2002.
- [11] Hidenori Takeda, Shoken M. Miyama, and Minoru Sekiya. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Progress of Theoretical Physics*, 92(5):939–960, 1994.
- [12] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.
- [13] Marco Sutti. SPH treatment of boundaries and application to moving objects. Technical report, École polytechnique fédérale de Lausanne, 2016.