

# Implementation of Unstructured Multi-dimensional THINC for Practical Multi-Phase Flow Simulations

Mohamed M. Kamra<sup>1,\*</sup>, Changhong Hu<sup>2</sup>

<sup>1</sup>Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Japan

<sup>2</sup>Research Institute of Applied Mechanics, Kyushu University, Japan

\*Author to whom correspondence should be addressed,

E-mail: mohamed.kamra@riam.kyushu-u.ac.jp

(Received December 15, 2016; accepted March 15, 2017).

This work presents an application of the Unstructured Multi-dimensional Tangent Hyperbolic Interface Capturing (UMTHINC), volume of fluid, scheme for the multi-phase solution of free surface flows on unstructured mesh. The applicability of the UMTHINC to practical engineering problems with interfacial multiphase flow is investigated and the accuracy of the results are reported. The UMTHINC is integrated into an in-house unstructured incompressible flow solver and used as an engine to capture the moving interface. The well-known dam break problems with and without an obstacle are used to evaluate the accuracy and performance of the scheme. This work is limited to two dimensional cases with no turbulence modeling. The relationship between the sharpness parameter  $\beta$ , and CFL number is examined and reported. The results are well analyzed and compared to experimental data wherever possible. The results show that the UMTHINC interface capturing method is able to accurately capture the interface without the complexity and computation cost of the geometrical reconstruction method.

Keywords: Multi-phase fluid dynamics; unstructured grid; interfacial flow.

## 1. Introduction

When considering flows with multiple fluids such as free surface flow and multiphase flow (droplets and bubbles), there are many approaches to solving such flows; the first and oldest approach is interface tracking, and the second is interface capturing. The latter is more popular with the CFD community due to its adaptability to different flow structures and different kinds of grids not to mention its computational efficiency.

The Tangent Hyperbolic Interface Capturing (THINC) method was originally developed by Xiao et al<sup>1)</sup> as an algebraic method of capturing the discontinuous transition of physical properties associated with multi-phase flows. Since the method provides an algebraic model for the transition between different fluids, it becomes relatively easy to compute the convective fluxes which is the target of advection schemes. The method was extended to unstructured grid by Satoshi Li et al.<sup>2,3)</sup> and Xie B. et al.<sup>4)</sup> and designated as Unstructured Multidimensional THINC (UMTHINC).

In this paper, the interface capturing volume of fluid method UMTHINC is applied to unstructured grid in two-dimensional free surface flow. The method is briefly described and the different numerical aspects of the methods are analyzed. The efficiency and accuracy of the method is investigated by solving the standard benchmark case problem of dam break with and without an obstacle.

## 2. Numerical Method

### 2.1 Transport equations

The two-dimensional incompressible Navier-Stokes equation is considered for the present simulations. No turbulence modeling was applied in this work and the surface tension effect was neglected due to its limited effect in such body force dominated flows. In order to facilitate the handling of pressure boundary conditions, the pressure and body force are modified as shown in eqn (1) to recover the simple zero gradient pressure boundary conditions.

$$\frac{DU}{Dt} = \nabla \cdot (\mu \nabla U) - \nabla p_d - \mathbf{g} \cdot \mathbf{x} \nabla \rho \quad (1)$$

where  $U$  is the velocity vector,  $\mu$  is kinematic viscosity,  $p_d$  is the working pressure variable,  $\mathbf{g}$  is the gravity acceleration vector and  $\rho$  is the density. The working pressure  $p_d$  is defined as  $p_d = p + \rho \mathbf{g} \cdot \mathbf{x}$  with  $p$  as the static pressure and  $\mathbf{x}$  as the position vector.

In this work, the volume of fluid method is used to model multiple fluids by considering an indicator/color function to distinguish between each fluid and the rest of the fluids. Such function  $C(\mathbf{x}, t)$  is defined such that it is equal to one if  $\mathbf{x}$  falls in fluid 1 and zero otherwise. The time evolution of the indicator/color function can be represented by the following advection equation

$$\frac{\partial C}{\partial t} + \nabla \cdot (U C) = C \nabla \cdot U \quad (2)$$

The volume fraction  $\phi$  in cell volume  $\Delta\Omega$  is defined as

$$\phi \equiv \frac{1}{\Delta\Omega_i} \int_{\Omega_i} C(x,t) dx \quad (3)$$

Rewriting equation (2) in term of the volume fraction

$$\frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{U} \phi) = \phi \nabla \cdot \mathbf{U} \quad (4)$$

The density and viscosity are computed from volume fraction as

$$\begin{aligned} \mu &= \mu_l \phi + \mu_g (1 - \phi) \\ \rho &= \rho_l \phi + \rho_g (1 - \phi) \end{aligned} \quad (5)$$

In the THINC method, the interface is defined as piecewise hyperbolic tangent profile, so the indicator/color function is now approximated as

$$C(\mathbf{x}) = \frac{1}{2} (1 + \tanh(\beta(P(\mathbf{x}) + d))) \quad (6)$$

where  $\beta$  is the sharpness parameter which controls how steep(thin) is the interface region between fluids,  $P(\mathbf{x}) + d = 0$  is the interface equation and  $P(x)$  can be a linear or quadratic polynomial with coefficients based on the unit normal vector  $(n_x, n_y)$  and curvature matrix  $I_{pq}$ ,  $p, q = x, y$ , of the interface. In this work only linear interface construction is considered. The reader is referred to references<sup>2-4</sup> for more details.

## 2.2 Numerical Algorithm

In this work, the two-dimensional incompressible laminar flow is solved using the non-iterative time advancement PISO method<sup>5-7</sup> with 3 neighbor correction and one non-orthogonally correction steps during each time step to ensure that both the continuity and momentum equations are satisfied.

First order time implicit scheme was applied for the time discretization. The second order linear upwind scheme was applied for the convection term and the least square method was used to compute the velocity gradients.

The finite volume formulation of the UMTINC method can be written as

$$\frac{\partial\phi_i(t)}{\partial t} + \frac{1}{\Delta\Omega_i} \sum_{j=1}^{N_{faces}} v_{n_{ij}} \phi_f = \frac{\phi_i(t)}{\Delta\Omega_i} \sum_{j=1}^{N_{faces}} v_{n_{ij}} \Delta S_{ij} \quad (7)$$

With the face value defined based on the THINC method approximation

$$\phi_f = \int_{S_{ij}} C_i(x) dS \quad (8)$$

The numerical method used to implement UMTINC can be summarized in the following steps:

- 1- The calculation of the interface unit normal  $n_x, n_y$
- 2- Calculation of the coefficient  $d$
- 3- Calculation of the face flux.
- 4- Advancing the volume fraction.

**Step1: The calculation of the interface normal  $n_x, n_y$**

The interface unit normal is computed from

$$\mathbf{n}_i = \frac{\nabla\phi_i}{\|\nabla\phi_i\|} \quad (9)$$

In reference<sup>4</sup>, the authors used the least square method to compute the gradient of the volume of fraction at the cell nodes/vertices then obtained the value at the cell center by simple averaging. We found such method to be computationally expensive, as it involves so many repeated memory visits, not to mention memory consuming. So other methods were attempted to calculate the gradient directly at the cell center. The cell-based least square method, LSQ, which computes the gradient based on the cell's nearest neighbors (cells that share a common face) only was chosen to compute the gradient.

**Step2: Calculation of the coefficient  $d$**

An approximation formula for the coefficient  $d$  is given by references<sup>2</sup>, the expression is derived by performing the analytical integration in the dominating direction (the direction with the highest absolute value of the normal component) and numerical integration (using Gauss-Legendre quadrature) in the other directions. In order to reduce the computational cost and obtain an algebraic expression for the coefficient  $d$ , only two points are used in the numerical integration. As a result, some error is to be expected in the value of  $d$  which means that

$$\phi_i - \frac{1}{\Delta\Omega_i} \int_{\Omega_i} C(x,t) dx = \varepsilon \quad (10)$$

Such error can cause instability and inaccuracy that would put some additional limitations to the maximum CFL number and maximum sharpness parameters that can be used. We found that this aspect requires more extensive study and investigation which is beyond the scope of this work.

**Step3: Calculation of the face flux**

Satoshi Li et al.<sup>2,3</sup> and Xie B. et al.<sup>4</sup> used Gauss-Legendre Quadrature to approximate the face flux across the face using 1 or 2 points. We investigated the effect of the number of points on the accuracy of the computations and found the effect to be quite small (less than 5%).

**Step4: Advancing the volume fraction.**

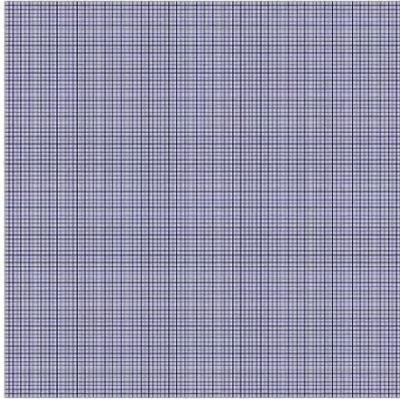
The explicit Euler and minimum storage explicit Runge-Kutta schemes, RK46NL<sup>8</sup>, was tested for time marching and no significant difference, other than the additional computational cost of the six stage scheme, was found. Consequently, the Euler explicit scheme was used for the time advancement of the volume fraction in this work.

## 3. Results

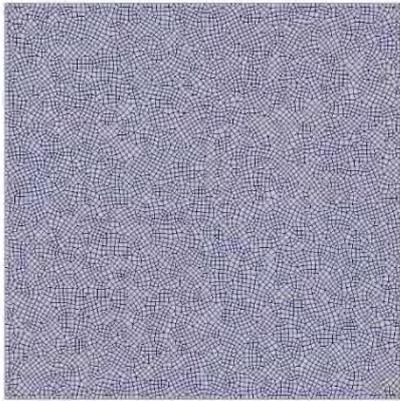
### 3.1 Dam break without an obstacle

The computational domain is taken as 1 m by 1 m with 100 cells in each dimension. Adaptive time stepping was adopted with a specified maximum value of CFL number and the simulation was conducted for 1.2 seconds. A sharpness parameter  $\beta$  of 3.0 was used for this simulation and interface. The water column is 0.2 m by 0.4 m with the density and viscosity of water taken as 997 kg/m<sup>3</sup> and 8.9x10<sup>-4</sup> m<sup>2</sup>/s respectively. The density and viscosity of air were taken as 1.185 kg/m<sup>3</sup> and 1.789x10<sup>-5</sup> m<sup>2</sup>/s and the gravity acceleration was set to 9.81 m/s<sup>2</sup>.

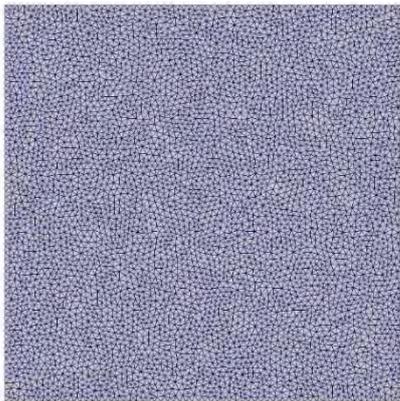
Three computational grids with different cell shapes were considered for this simulation and designated as A, B, and C as shown in Fig. 1. The number of cells for all three grids is roughly 10,000 cells with an average (effective) cell size of 0.01m.



(a) Grid A



(b) Grid B



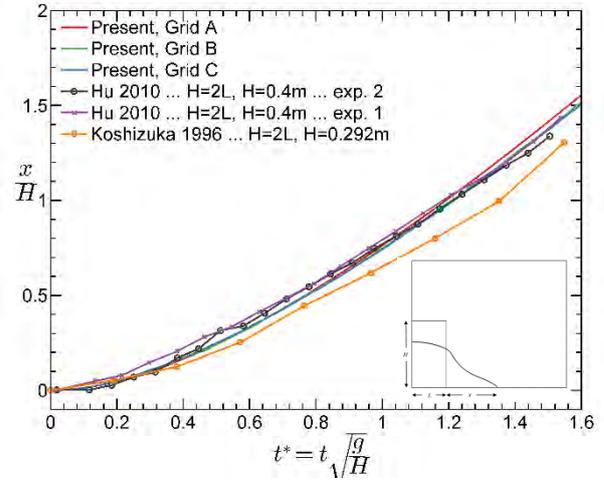
(c) Grid C

**Fig. 1:** Computational grid in dam break case without an obstacle

The results were compared to experimental data of Hu and Seyoushi<sup>9)</sup> and Koshizuka and Oka<sup>10)</sup> as shown in Fig 2.

The computed results on all considered grids are consistent with each other and compare very well with Hu and Seyoushi<sup>9)</sup> but with some deviation with Koshizuka<sup>10)</sup>.

The deviation may be attributed to the difference in the water column height or the experimental setup such as gate motion<sup>11)</sup>.



**Fig. 2:** Comparison of the wave front speed with experimental data<sup>9,10)</sup>

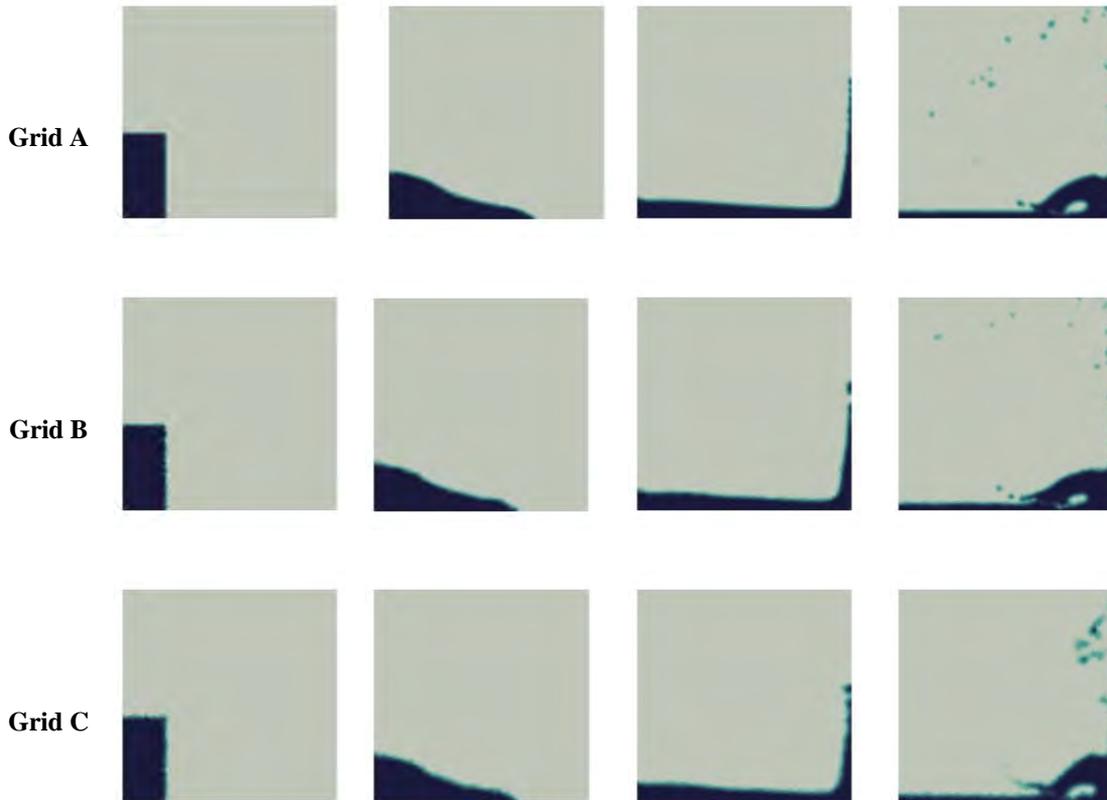
An investigation into the relation between the sharpness parameter  $\beta$  and the max CFL number was conducted and the following findings were obtained. A sharpness parameter  $\beta$  between 2.5 and 4.0 is necessary to accurately resolve the interface within one or two cells at best. For this range, the maximum CFL number necessary to maintain a stable and bounded solution is 0.20, and 0.25 for triangular and quad mesh respectively. These findings are consistent with limits provided by Satoshi Li et al.<sup>3)</sup> for triangular and tetrahedral mesh. Exceeding these limits may result in unbounded values and eventually compromising the stability of the solver. Such behavior may be attributed to the inaccuracy in computing the coefficient value  $d$ , as mentioned in the previous section, or the inaccuracy of the interface normal values.

For the remainder of this work, the max. CFL number used to produce the following results is taken as 0.15 regardless of the mesh type.

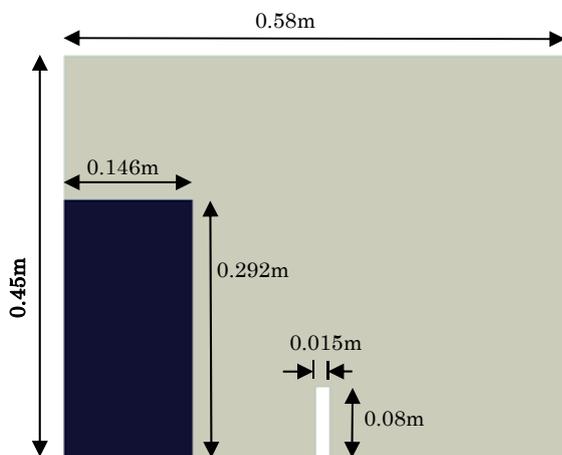
Fig. 3 depicts a comparison between the free surface profile on the three grids employed in the case. The comparison shows that the computed solutions are fairly similar with some differences on grid C at  $t = 1.0s$ . This can be attributed to the poor behavior of triangular mesh near the walls, especially on such coarse mesh. Another point, which can also be seen in Fig 2, is that wave front on structured quad mesh is a bit faster than the other grids. A reason for this could be the inaccurate boundary layer treatment or insufficient orthogonality correction.

### 3.2 Dam break with an obstacle

The dam break with an obstacle was solved using the same solver configuration but with different grids. The initial configuration of the problem is shown in Fig. 4. The first grid is a uniform quad mesh with an average cell size of 0.005 m.



**Fig. 3:** Evolution of the free surface profile on three different grids for the dam break without an obstacle at time instances (from left to right)  $t = 0.0, 0.25, 0.50,$  and  $1.0$  seconds

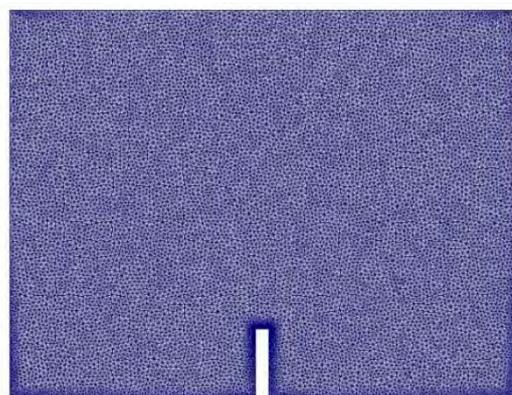


**Fig. 4:** Dam break with an obstacle (Initial configuration)

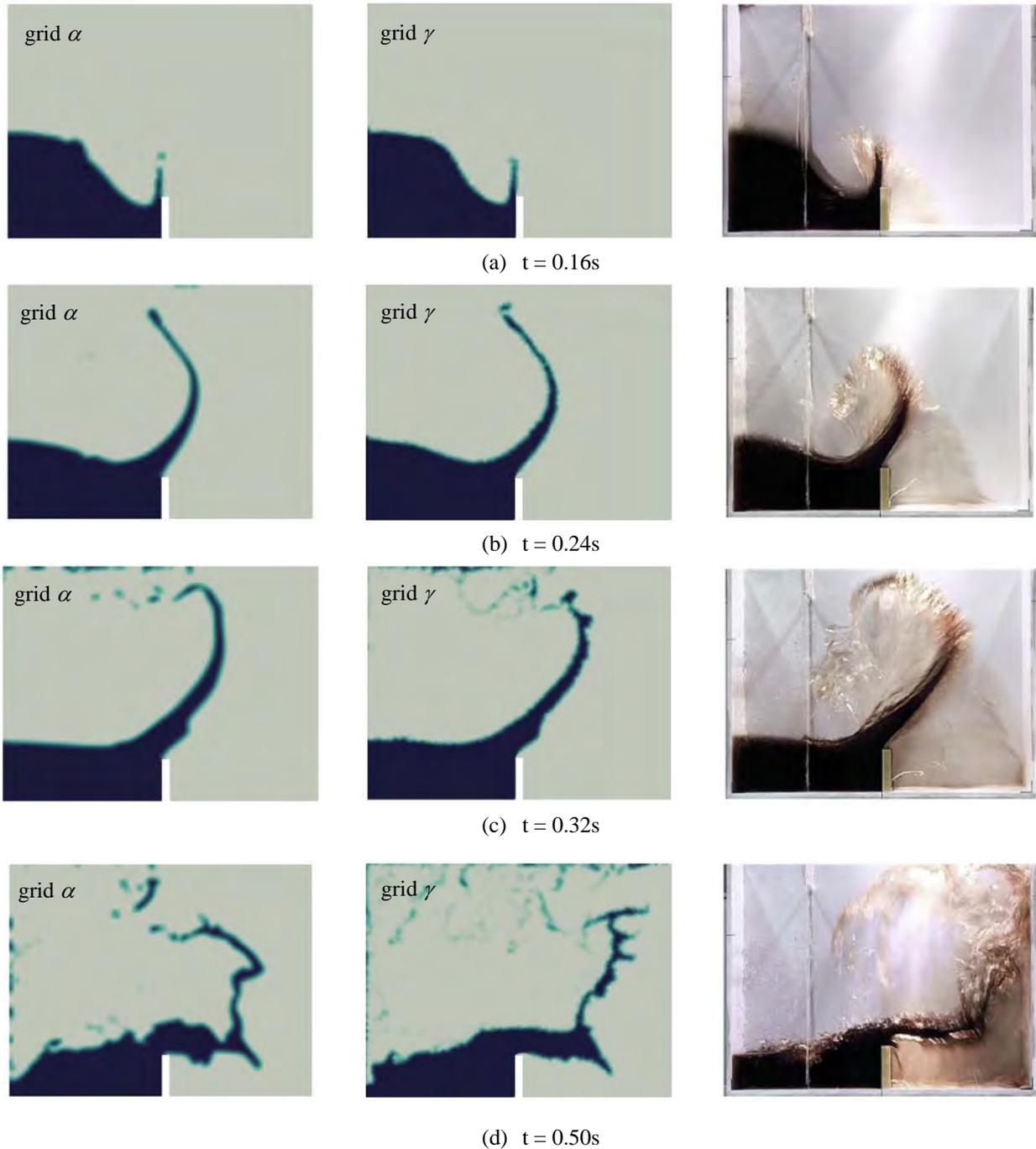
The second is a triangular mesh (as shown in Fig. 5), which was refined near the wall boundaries, with an average cell size of 0.002 m. The two grids will be designated as grid  $\alpha$  and grid  $\gamma$ .

The computed results are compared with the experimental data provided by Kolke<sup>12</sup>. For this problem, the comparison is qualitative and limited to the visualization of the free surface profile at specified times. Referring to Fig 6, the computed results compare well with

the experimental data on both grids with some differences observed at  $t = 0.50$  s. An interesting point is that the computed solution on grid  $\gamma$  appears to agree with the experiment much better than grid  $\alpha$ . This agreement could be the result of the excessive cell refinement near the walls and especially the obstacle walls. Another interesting observation is the loss of momentum in the computed solution at  $t = 0.5$ s as the water, unlike the experiment, doesn't reach the downstream wall. The reason for this could be the insufficient accuracy of the momentum advection scheme and omission of the surface tension contribution which affects thin splash waves.



**Fig. 5:** Computational grid (grid  $\gamma$ ) in dam break case with an obstacle



**Fig. 6:** Comparison of present numerical results with experimental data<sup>12</sup>

#### 4. Conclusion

The UMTHINC method was applied to solve the violent impact of water resulted from dam break with sufficient accuracy when compared to experimental data. The method was found capable of handling both unstructured quad and triangular mesh with no additional complication in coding. The triangle mesh, however, was found to require further refinement near the boundary walls and in areas with expected presence of small droplets and bubbles in order to accurately resolve the interface.

A relatively low courant number ( $CFL < 0.25$ ) was found necessary to ensure an accurate and bounded solution of the volume fraction at a sufficiently high sharpness parameter ( $\beta > 2.5$ ).

Extensive study is necessary to determine the extent effect of the inaccuracy in calculating the coefficient  $d$  on the overall accuracy, boundedness, and stability of the solution.

## References

- 1) Xiao F, Honma Y, Kono T. A simple algebraic interface capturing scheme using hyperbolic tangent function. *Int J Numer Methods Fluids*, **48**(9):1023-1040, 2005.
- 2) Li S, Sugiyama K, Takeuchi S, Takagi S, Matsumoto Y, Xiao F. An interface capturing method with a continuous function: The THINC method with multi-dimensional reconstruction. *J Comput Phys*, **231**(5):2328-2358, 2012.
- 3) Li S, Xie B, Xiao F. An interface capturing method with a continuous function: The THINC method on unstructured triangular and tetrahedral meshes. *J Comput Phys*, **259**:260-269, 2014.
- 4) Xie B, Li S, Xiao F. An efficient and accurate algebraic interface capturing method for unstructured grids in 2 and 3 dimensions: The THINC method with quadratic surface representation. *Int J Numer Methods Fluids*, **76**(12):1025-1042, 2014.
- 5) Versteeg HK, Malalasekera W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Ltd.; 2007.
- 6) Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg; 2012.
- 7) Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. 1997.
- 8) Berland J, Bogey C, Bailly C. Low-dissipation and low-dispersion fourth-order Runge–Kutta algorithm. *Comput Fluids*, **35**(10):1459–1463, 2006.
- 9) Hu C, Sueyoshi M. Numerical simulation and experiment on dam break problem. *J Mar Sci Appl.*, **9**(2):109-114, 2010.
- 10) Koshizuka S, Oka Y. *Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid*, Vol **123**. American Nuclear Society; 1996.
- 11) Ye Z, Zhao X, Deng Z. Numerical investigation of the gate motion effect on a dam break flow. *J Mar Sci Technol.*, **21**(4):579-591, 2016.
- 12) Kölke A. Modellierung und Diskretisierung bewegter Diskontinuitäten in randgekoppelten Mehrfeldsystemen. 2005.