Title: Numerically Modelling 2D Swash Zone Flows: SMAC VS VOF

Author: Anton Ali[1*]

[1] Lecturer - Department of Civil & Environmental Engineering, The University of the West Indies, St. Augustine campus, Trinidad and Tobago

[*]Corresponding author

Corresponding author email: anton.ali@uwi.edu

Word Count: 3474

Abstract

In recent years, numerically modelling the swash-zone has proven a feasible approach in unraveling the zone's intricacies. Whilst many researchers have used and examined many swash-zone numerical models, no specific research has examined the comparative performance of the Simplified Marker and Cell (SMAC) method relative to the status quo; the Volume of Fluid (VOF) method, in 2D applications of the zone. This study investigates the performance of these numerical methods via comparative simulations of 2D dam-break swash like flows across both models. The results show that an adapted Finite Difference (FD)- Finite Volume (FV) discretization of the SMAC method with reductions in marker particles yields smooth interfacial predictions and clear convergence with mesh refinement, while being less computationally demanding than the VOF method for the test scenario. The study demonstrates the suitability of the SMAC method for simulating swash zone flows and highlights its potential for efficient and accurate 2D simulations.

Keywords: Computational Fluid Dynamics (CFD); MATALAB; Numerical Model; Simplified Marker and Cell (SMAC); Swash Zone; Volume of Fluid (VOF).

1. Introduction

The swash zone is a complex temporal region encompassing the moving shoreline. The water depths are small, and the flows are incompressible and Newtonian in the zone, outlining the suitability of the Navier-Stokes Equations (NSE) (Briganti et al. 2016). The earliest implementation of the NSE in the zone dates to 1984, for turbulent bore simulations on a beach by Svendsen and Madsen (Mory et al. 2011). Presently, such depth-resolving models have contributed to great advancement of turbulence, bore collapse and infiltration/exfiltration effects within the zone. However, the marked improvements in accuracy and flow detailing facilitated by such models are also accompanied with a significant increase in computational cost (Briganti et al. 2016). More specifically, the increased computational cost is directly linked to added numerical computations arising from the complexity of solving the complete vertical flow structures of the entire fluid field (Kranenborg et al. 2019). This single drawback is often considered the major deterrent of applying such models within the zone.

Additionally, in applying the NSE to the sub-aerial region of the zone, the added complexity of multiphase flows must also be incorporated due to the transient nature of the swash lens (Ali and Villarroel-Lamb 2020). In general, solving the NSE is very complex such that analytical solutions exist for only a few simple investigations (Krüger et al. 2017). Moreover, multiphase flows present a challenging subdivision due to the demands of moving interfaces, material property discontinuities and complex flow fields near interfaces (Yeoh and Tu 2010). This study investigates the performance of two such multiphase models: (1) the Simplified Marker and Cell (SMAC) method and (2) the Volume of Fluid (VOF) method in a representative two -dimensional (2D) swash zone application via MATLAB.

2. Relevant background

Depth-resolving models of the swash zone employ some version of the NSE (equations 1 – 2). For most applications, these equations may be simplified by the assumption of constant density and the approximation of a Newtonian stress tensor for the viscous terms. Although such simplifications are possible, the solutions to variations of equations (1) and (2) are so complex that numerical approximations are widely pursued. The general nature of which forms the sub-field of Computational Fluid Dynamics (CFD).

$$\partial\rho/\partial t + \nabla \cdot (\rho \vec{V}) = 0 \tag{1}$$

$$\partial\rho\vec{V}/\partial t + \nabla \cdot (\rho\vec{V}\vec{V}) = -\nabla P + \nabla \cdot (\tau) + \rho g + F_{Ex} \tag{2}$$

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

Where: P, ρ and τ are the fluid pressure, density and viscous stress tensor respectively, $\overrightarrow{V}$ is the fluid velocity vector, g is the acceleration due to gravity and F_{Ex} is any other external forces acting on the fluid.

Conventional CFD utilizes either the Finite Difference Method (FD) method, Finite Volume (FV) method or Finite Element (FE) method in representing the NSE under a discretization approach. Added complexities such as closure models for multiphase and turbulence considerations may also be appended based on the physical nature of the investigation.

2.1 Simplified Marker and Cell (SMAC) method

The SMAC method is an improvement of the early pioneering Eulerian interface evolution Marker and Cell (MAC) model developed by Harlow and Welch (1965) specific to the modelling of 2D incompressible multiphase flows. The SMAC method utilizes a fixed Eulerian computational domain (Figure 1) to solve fluid velocities via equations (1) and (2) (Yeoh and Tu 2010), whilst the advection of “massless” marker particles details the evolution of fluid phases and interfaces (Johnson 1996). The method classifies the Eulerian mesh elements based on the presence of marker particles in mech cells (Figure 1). Empty cells are assigned with no marker particles whilst both full and surface cells contain particles. Distinctions between full and surface cells are based on cell neighbours. Specifically, cells that contain particles and are also adjacent to at least one empty cell are classified as surface cells, whilst full cells possess no empty neighbours (Tomé and McKee 1994). The method employs a cell reflagging procedure after marker particle advection which details fluid evolution.

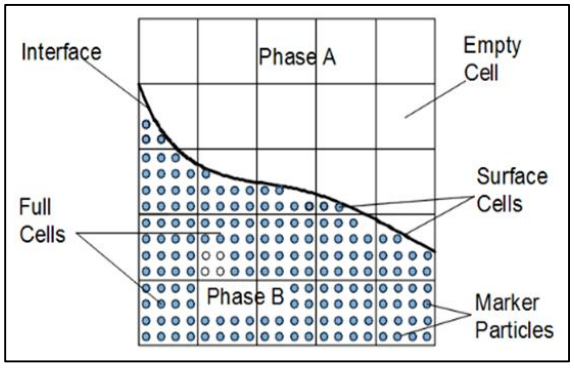

Figure 1: Typical MAC configuration

The improvement of SMAC compared to the original MAC method is the use of a split calculation cycle. This cycle involves the computation of a tentative velocity field in the first step,

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

followed by a correction via a potential function that satisfies incompressibility in the second step (Amsden and Harlow 1970).

2.2 Volume of Fluid (VOF) method

The volume of fluid (VOF) method was developed by Hirt and Nichols in 1981 as a successor to the original MAC method and is one of the most common multiphase models used presently. The method also utilizes a fixed Eulerian computational domain (Figure 2) to solve fluid velocities via equations (1) and (2). However, the VOF method utilizes an interface capturing method via an indicator function to identify fluid phases and a Lagrangian invariant statement for the evolution of fluid phases (Prosperetti and Tryggvason 2007). The method terms the indicator function as the volume fraction function and computes its value as the volume ratio of a selected fluid phase to the volume of the computational cell within which it is present. In the case of two fluid phases (Figure 2), where phase A is utilized as the referenced phase, a cell consisting entirely of phase A has a volume fraction value of 1, a cell entirely filled with phase B has a value of 0 and values ranging between 0 & 1 contain both phase A & B being representative of an interface cell (Daftari 2010).

The VOF can utilize both the sharp and the immersed interface approach. In the pioneering publication by Hirt and Nichols (1981), the sharp approach was used. However, the commonly used VOF method solves the fluid in the entire domain (the homogenous one fluid) along with an interface capturing and reconstruction scheme, under the immersed interface approach (Gao and Sun 2011; Gao, Morley and Dhir 2003). For the commonly used VOF immersed interface approach, an interface reconstruction is done that approximates the interface from the volume fraction field along with the inclusion of an advection procedure which transports the volume function domain information with time (Puckett et al. 1997).

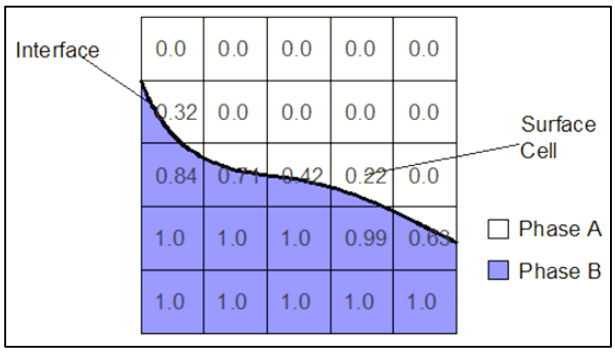

Figure 2: Typical VOF configuration

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

The advection procedure can adopt either of two approaches: a split or unsplit approach. A variety of procedures can be used for interface reconstruction, falling into either donor and acceptor formulations or geometric reconstruction (Yeoh and Tu 2010). Donor-Acceptor formulations utilize volume fraction field information upstream and downstream of a reference cell as a basis for interface reconstruction (Yeoh and Tu 2010). The geometric reconstruction approach represents interfaces as line segments within a computational cell (Yeoh and Tu 2010). Either Simple Line Interface Calculation (SLIC) or Piecewise Liner Interface Calculation (PLIC) methods are available under this approach. Reference is made to relevant literature such as Prosperetti & Tryggvason (2007) or Yeoh and Tu (2010) for further information regarding SLIC and PLIC.

3. Materials and Methodology

3.1 Test Scenario

A 2D dam-break test case was used in this study. Its set-up was achieved via sectioning an area of an overall region and building a liquid-fluid reservoir at one end for a chosen height, H and length, L (Figure 3). The reservoir was confined by three bounding walls (Wall A, Wall D and Wall B). The instantaneous removal of wall D creates the test problem whilst walls A-C were fixed. The reservoir used in the scenario is of dimensions 7.5cm x 5cm (H x L), comprising water within an overall domain region where the horizontal distance between wall A and C is 35cm. Moreover, the test was chosen since a dam-break scenario has been shown to be hydrodynamically similar to swash flows (Peregrine and Williams 2001).

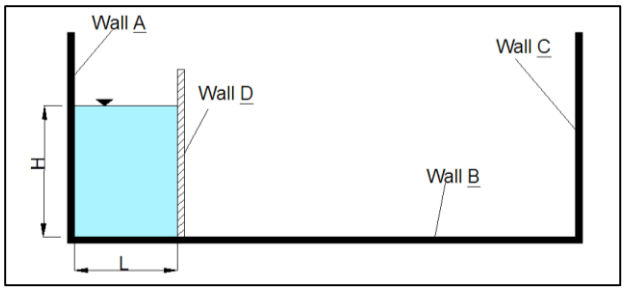

Figure 3: 2D dam-break test scenario

ISSN: 3050-9696 E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

3.2 SMAC Test Scenario Set-Up

3.2.1 SMAC Algorithm Implementation

The SMAC method utilizes a staggered grid where fluid velocities are referenced at cell faces whilst pressures are located at cell centers. It is well known that a strong coupling between the fluid pressure and velocity fields is obtained via such a grid (Tu, Inthavong and Ahmadi 2013).

The method details fluid evolution by advecting virtual particles for computed velocity fields on such grids. For the 2D test problem, equations (1) and (2) are given by expanded equations (3), (4) and (5) as adapted from Tomé and McKee (1994).

$$\partial u/\partial x + \partial v/\partial y = 0 \tag{3}$$

$$\partial u/\partial t + \partial u^2/\partial x + \partial uv/\partial y = -1/\rho \cdot \partial P/\partial x + \mu/\rho \cdot \partial/\partial y\left(\partial u/\partial y - \partial v/\partial x\right) + g_x \tag{4}$$

$$\partial v/\partial t + \partial uv/\partial x + \partial v^2/\partial y = -1/\rho \cdot \partial P/\partial y - \mu/\rho \cdot \partial/\partial y\left(\partial u/\partial y - \partial v/\partial x\right) + g_y \tag{5}$$

Where: u is the x-component of velocity, v is the y-component of velocity and g_x and g_y are the x-component and y-components of the acceleration due to gravity.

The general SMAC procedure, adapted from Amsden and Harlow (1970), Tomé and McKee (1994) and Tomé et al. (2000) is as follows. Assuming the initial velocity field, $U(x, t_0)$ is known at time t_0 and the relevant boundary conditions are known. The updated velocity field at time $t = t_0 + \Delta t$ can be computed:

1. Let $\tilde{P}(x, t_0)$ be an arbitrary pressure field which satisfies the correct boundary conditions at the interface.
2. Compute the intermediate velocity fields $\tilde{U}(x, t_0)$, from:

$$\partial \tilde{u}/\partial t = \left[-\partial u^2/\partial x - \partial uv/\partial y - 1/\rho \cdot \partial \tilde{P}/\partial x + \mu/\rho \cdot \partial/\partial y\left(\partial u/\partial y - \partial v/\partial x\right) + g_x\right]_{t=t_0} \tag{6}$$

$$\partial \tilde{v}/\partial t = \left[-\partial uv/\partial x - \partial v^2/\partial y - 1/\rho \cdot \partial \tilde{P}/\partial y - \mu/\rho \cdot \partial/\partial x\left(\partial u/\partial y - \partial v/\partial x\right) + g_y\right]_{t=t_0} \tag{7}$$

ISSN: 3050-9696 E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

With $\tilde{U}(x, t_0) = U(x, t_0)$ using correct boundary conditions. Tomé, Duffy and McKee (1996) outlined that $\tilde{U}(x,t)$ satisfies the correct vorticity but not equation (2.24). Thus, corrections are needed where the true velocity field is now defined as:

$$U(x,t) = \tilde{U}(x,t) - \nabla\Psi(x,t) \tag{8}$$

$$\nabla^2\Psi = \nabla \cdot \tilde{U}(x,t) \tag{9}$$

So that U(x,t) satisfies equation (3) as well as vorticity. Where Ψ is the velocity potential function.

3. Solve the Poisson equation:

$$\nabla^2\Psi = \nabla \cdot \tilde{U}(x,t) \tag{10}$$

4. Compute the velocity field:

$$U(x,t) = \tilde{U}(x,t) - \nabla\Psi(x,t) \tag{11}$$

5. Compute the pressure field, as outlined by Tomé, Duffy and McKee (1996):

$$P(x,t) = \tilde{P}(x,t) + {}^{\Psi(x,t)}/_{\delta t} \tag{12}$$

6. Advect the marker particles for the computed velocity field in step (4) via:

$${}^{\partial x}/_{\partial t} = u, \quad {}^{\partial y}/_{\partial t} = v \tag{13}$$

7. Reflagging of the domain and adjusting U(x,t) to satisfy equation (3) for newly created computational fluid cells.

The beforementioned method was implemented in MATLAB via a FDM discretization for all terms except the convective terms which were discretized via a weighted average FVM scheme. Reference is made to Kothe, Mjolsness and Torey (1991) for further details. The discretized equations solve equations (6) and (7) explicitly via a forward differencing in time, whilst a sparse symmetric system (equation (10)) was used to acquire values for the potential function. The methodology outlined above utilized the standard projection method which consists of two main parts. The first is commonly termed the predictor step and involves the computation of an intermediate velocity field, "$\tilde{U}$"(x, t). This is done via explicitly marching forward in time utilizing any arbitrary pressure field which incorporates the boundary conditions specific to the test problem; i.e. Step (2). The second part of the projection method is a corrector step, commonly known as the projection step. In this part, "$\tilde{U}$"(x, t) was corrected to the divergence-free velocity field, U(x,t) by the addition of a velocity potential Ψ(x,t). This velocity potential is acquired via the

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

Poisson Pressure Equation (PPE) given by equation (10). These values were then used to update marker particle locations, hence modelling flow progression.

Updated particle locations can result in particles moving to new mesh cells which may not have already contained marker particles, for previously vacant cells which now contain marker particles, a cell-specific adjustment of U(x,t) is required to satisfy equation (3). Similar adjustments are also required for previously filled mech cells which are now void of particles. This classification is updated at each time increment and classifies cells as Boundary cells (B), Empty cells (E), Full cells (F) and Surface cells (S) as illustrated in Figure 4.

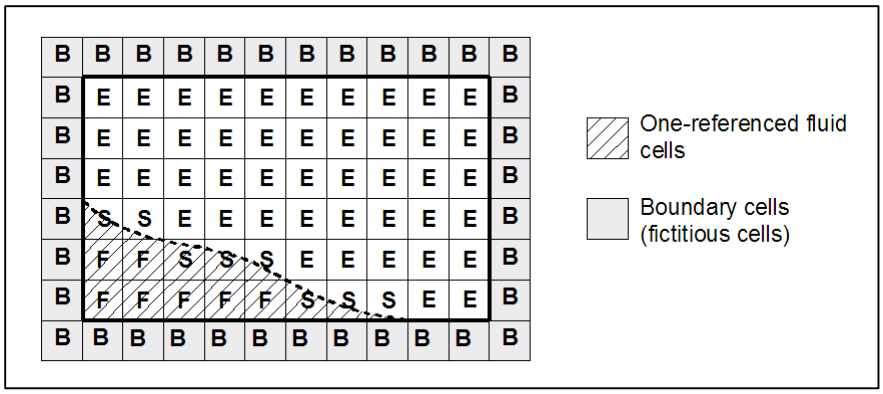

Figure 4: SMAC cell classification

3.2.2 SMAC Dam-Break MATLAB Set-Up

In SMAC algorithm implemented in MATLAB, four (4) uniform grid meshes of cell side sizes 0.5cm, 0.25cm, 0.1cm and 0.0625cm were built. Boundary conditions were set-up such that the right, bottom and left boundaries were solid no-slip walls, whilst the top boundary was set as a pressure outlet. Only one-fluid material was needed, this being water-liquid with $\rho = 0.997$ g/cm3 and $\mu = 8.9 \times 10^{-3}$ g/cms. A first-order upwind scheme was chosen for the representation of convective terms. Gravity was also selected as -980.6 cm/s2 and simulation was run at a fixed timestep of 1×10^{-4} s. The dam reservoir location was then defined in the domain for the 7.5cm x 5cm dimensions via marker particles solely at interfacial cells (approx. 50 marker particles per computational interfacial cell). Figure 5 shows the initial model set-up for 0.5cm x 0.5cm mesh grid. The bottom right hand of the mesh was taken as at x=0, y=0 and fictitious boundary cells are not shown. The solution was permitted to calculate until simulation results depicted contact with the right wall boundary (wall C - Figure 3).

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

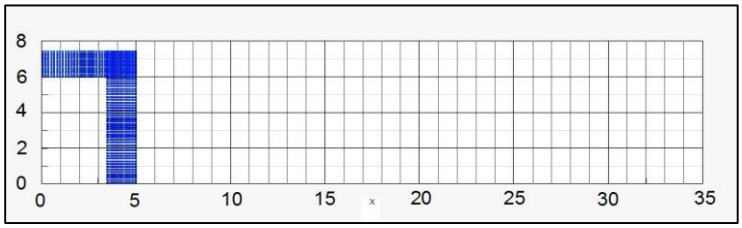

Figure 5: Dam-break set-up for SMAC method implemented in MATLAB for a 0.5cm x 0.5cm mesh grid at time = 0.0s

3.3 VOF Scenario Set-Up

3.3.1 VOF Algorithm Implementation

The VOF method implemented in MATALB was set-up similar to the SMAC method algorithm in MATLAB. The overlapping sections with the SMAC method, i.e., the FDM- FVM discretization in steps 1 – 5, were implemented congruently to the SMAC method. Further details for this implementation can be found in Hirt and Nichols (1981) and Kothe, Mjolsness and Torrey (1991). Sections unique to the VOF framework, namely pertaining to the advection of the volume fraction indicator function, adopted an interface evolution procedure according to the Young (1982). The procedure followed a piecewise linear approach, which is analogous to the geometric reconstruction scheme used in many commercial software (e.g. ANSYS Fluent 17.2). Under this approach, a split avenue as outlined by Pillord and Puckett (2004) was chosen for code implementation. Further details regarding alternate interface evolution avenue/approaches can be found in Rudman (1997), Pillord and Puckett (2004) and May et al. (2011).

MATLAB implementation equity between the VOF and SMAC method was followed by utilizing the same sub-routine codes where possible. Both methods also adopted similar strategies of exploiting the matrix/array nature of MATLAB when possible. “For-loops” were avoided where possible and all calculations and necessary matrix assemblies were done via logical arguments and matrix manipulations in MATLAB.

3.3.2 VOF Dam-Break MATLAB Set-Up

Four (4) uniform grid meshes of cell side sizes 0.5cm, 0.25cm, 0.1cm and 0.0625cm were also built in the VOF MATLAB algorithm. Boundary conditions were set-up similar to the SMAC scenario. Two fluid phases were set-up, air and water-liquid respectively. The properties for air were defined as $\rho = 1.225 \times 10^{-3}$ g/cm3 and $\mu = 1.837 \times 10^{-4}$ g/cms, whilst for water-liquid, $\rho =$ 0.997 g/cm3 and $\mu = 8.9 \times 10^{-3}$ g/cms. The dam reservoir location was then defined in the domain and operating conditions were set for gravity = -980.6 cm/s2. A first-order upwind scheme was

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

selected for the convective terms. The dam reservoir location was then defined in the domain for the 7.5cm x 5cm dimensions via the indicator function where the fluid interface stretched across one computational cell. Figure 6 shows the initial model set-up for a 0.5cm x 0.5cm mesh grid. Similar to the SMAC scenario, the solution was permitted to calculate until simulation results depicted contact with the right wall boundary.

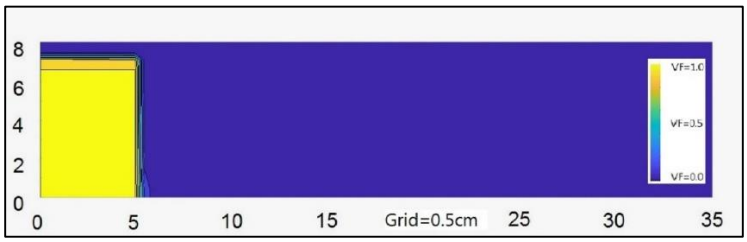

Figure 6: Dam-break set-up for VOF method implemented in MATLAB for a 0.5cm x 0.5cm mesh grid at time = 0.0s

MATLAB implementation equity between the VOF and SMAC method was followed by utilizing the same sub-routine codes where applicable. Both methods also adopted similar strategies of exploiting the matrix/array nature of MATLAB. “For-loops” were avoided, and all calculations and necessary matrix assemblies were done via logical arguments and matrix manipulations in MATLAB. Structuring both codes in such a manner allowed many areas of both methods to be coded similarly, where differences were only evident in areas unique to each multiphase model.

4. Results and Analysis

Comparative representative results between the SMAC and VOF implementations in MATLAB are presented in Figures 7 – 9. The results detail simulation predictions across the grid mesh sizes 0.25cm and 0.10cm, along with average computational time per simulation time-step across all the meshes. The simulation predictions outline marked differences between the VOF and SMAC methods at later time steps for the coarser 0.25cm mesh grid. Specifically, at times 0.15 and 0.25s the differences between the leading water edge locations between both methods are ≈1.5cm and ≈3.0cm respectively. However, as the mesh grid size decreased to the 0.1cm grid, these differences also decreased to ≈0.25cm and ≈0.75cm respectively. Thus, as mesh grid size resolution decreases, the methods appear to converge towards the same solution.

A more in-depth analysis of the predictions across Figures 7 and 8 outlines that as mesh grid resolution decreases both solutions appear to approach the exact solution for the leading water

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

edge from differing directions. This is evident at computational time 0.25s where the VOF method predicts a leading water edge location of ≈31.25cm for the 0.25cm mesh grid which increases to ≈34.75cm for the 0.1cm mesh grid. Conversely, at the same computational time of 0.25s, the SMAC method predicts a leading water edge location of ≈34.50cm for the 0.25cm mesh grid which decreases to ≈34.00cm for the 0.1cm mesh grid.

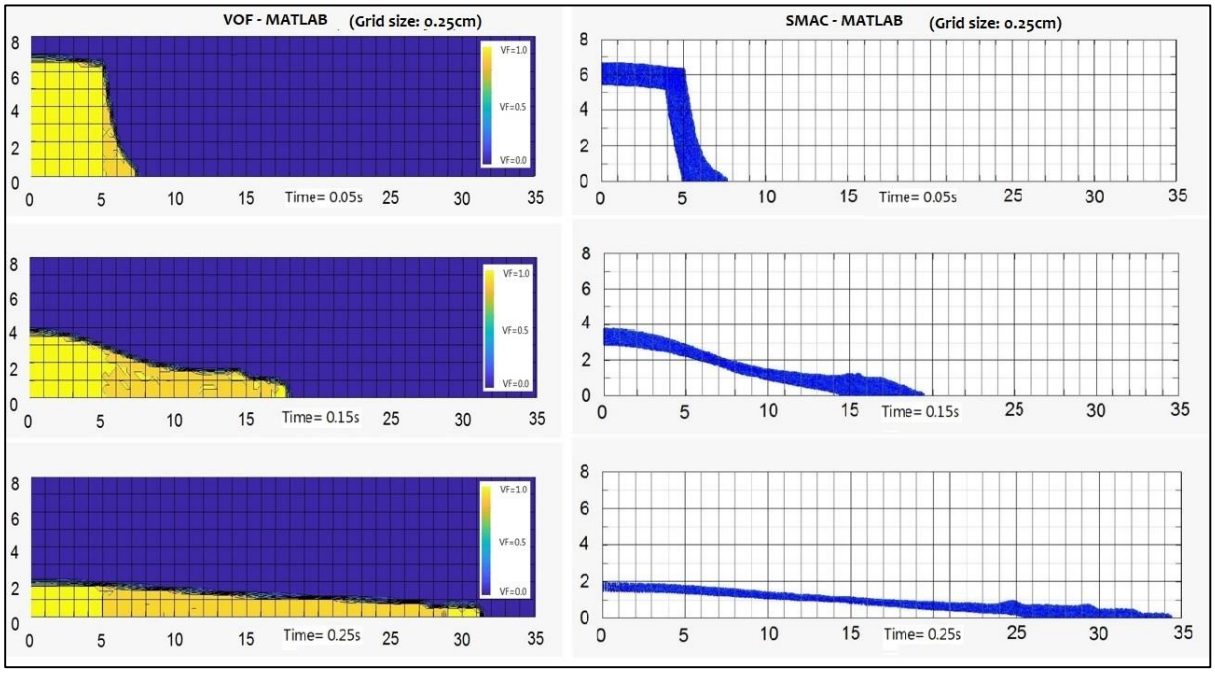

Figure 7: Dam-break results for VOF MATLAB and SMAC MATLAB for a 0.25cm x 0.25cm mesh grid at time = 0.05s, 0.15s and 0.25s

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

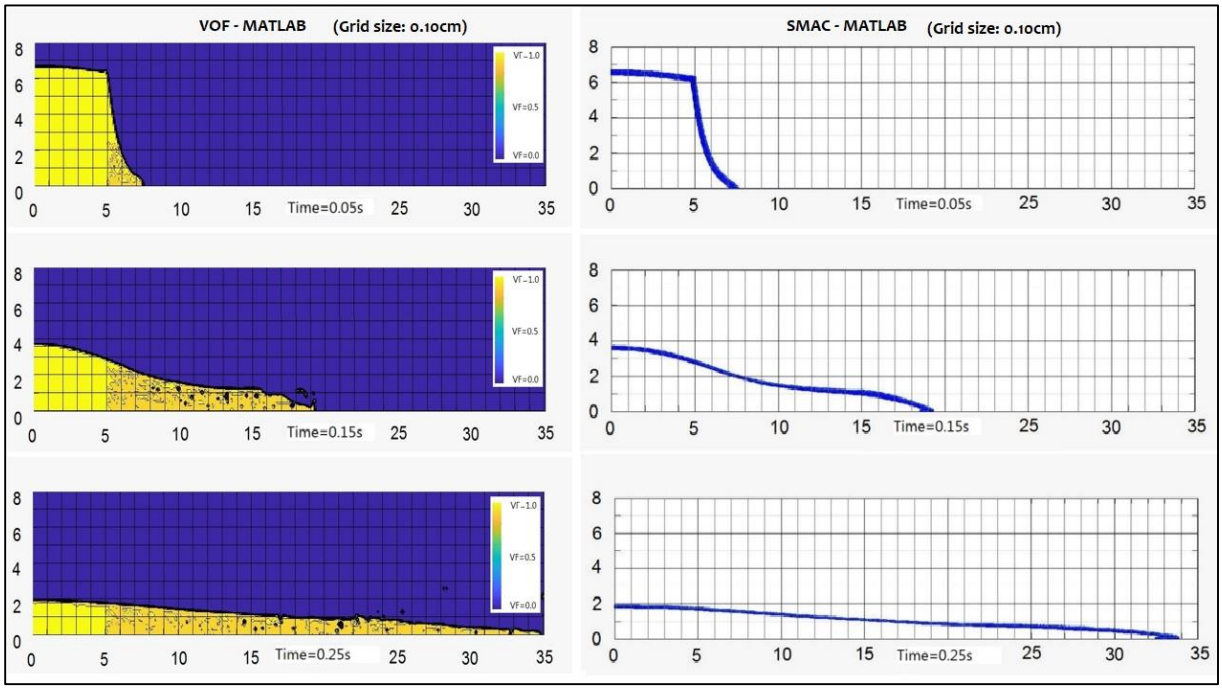

Figure 8: Dam-break results for VOF MATLAB and SMAC MATLAB for a 0.10cm x 0.10cm mesh grid at time = 0.05s, 0.15s and 0.25s

Analyzing the SMAC results, noteworthy interfacial oscillations are also seen to be reduced as the mesh grid size also decreases. This observation is noted across Figures 7 and 8, outlining that smooth and sharp interfaces are acquired with mesh resolution increases. Regarding interface predictions by the VOF method, the results highlight the generation of numerical errors as grid size decreased. This is seen specifically in Figure 8 for the 0.1cm grid where pockets of erroneous volume fractions are predicted. The absence of numerical filters and interface smoothing, lack of more complex descriptions of Young's fluxes, as well as, the use of a split advection method may be possible reasons for this observation.

Considering the average computational times for the SMAC- MATLAB and VOF- MATLAB methods presented in Figure 9. These observations are acquired from the dam-break test problem solved upon the four (4) mesh grid sizes considered and do not incorporate pre-processing activities such as mesh generation and solution initialization but rather reflect the raw computing time for the average calculation cycle. Comparisons made solely on the calculation cycle ensure no unfair bias is given to one particular method. The times presented

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

were all acquired from the computing device with 2019a academic version of the MATLAB platform and executed upon a standard Dell 64-bit desktop with intel core i7- 7700 HQ CPU @ 2.80Ghz with 16 GB of RAM.

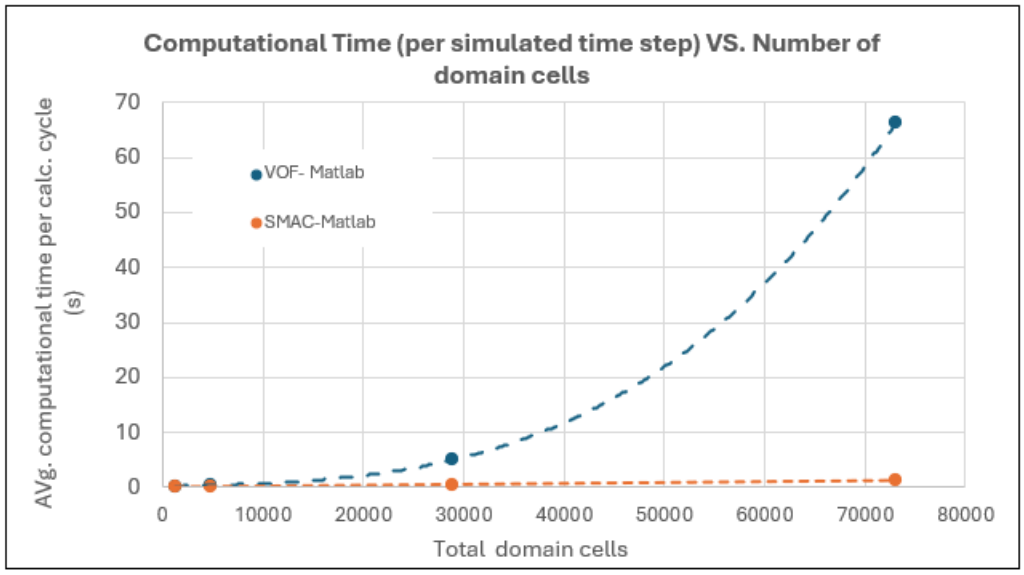

Figure 9: Computational time plot comparison between VOF- MATLAB and SMAC - MATLAB

The results outline that the SMAC method is less computationally expensive than the continuum VOF method for a swash like flow. This is most observable as grid size decreases and the number of domain cells increases. For instance, when the 0.0625cm grid (73060 mesh cells) is used, the VOF method requires an average of 66.7 seconds per calculation cycle whilst SMAC requires 1.2s. This is accredited to the fact that all of the 73060 mesh cells are used throughout the entire simulation (at all times) in the VOF model. Conversely, a variable percentage of the total 73060 mesh cells is used by the one-referenced fluid SMAC method; where the exact number of mesh cells varies at different times during the simulation due to interface evolution. It is unclear at what number of domain cells this linear SMAC variation may change. It should also be noted that an examination of the code run-times outlined that the sections of the continuum one-fluid VOF code which were most computationally demanding was the solution of the PPE equation, as well as, reconstruction and advection of the VOF function

Admittedly, the research does not consider the impacts of other relevant closure models, such as surface tension and turbulence, which may be necessary in the description of swash zone flows involving breaking waves and bore collapse. The impacts of such models on both SMAC

and VOF model predictions, interfacial mapping and computational cost remain the subject of further research. Still, the current findings outline the SMAC method as a viable numerical technique for detailing swash zone flows.

5. Conclusion

The study demonstrates the suitability of the combined finite difference-finite volume discretization of the SMAC method for the numerical description of 2D swash zone flows. The method's ability to mimic swash-like flows, yield noteworthy observations of smooth interfacial predictions and clear convergence with reductions in mesh resolution underline the mentioned. Further, based on the results, it can be concluded that the SMAC method with the improvements of a split calculation cycle and limitation of marker particles to interfacial cells only, is less computationally demanding than the commonly used continuum one-fluid VOF method for the scenarios considered. Further research into the performance of the method in the swash zone, with specific considerations of surface tension effects and the inclusion of turbulence variations appears appealing.

Acknowledgements

The author wishes to express his gratitude to the staff of the Civil Engineering Department at the University of the West Indies for their guidance throughout this study. Special thanks are also extended to the Computer Laboratory personnel, Dr. Varuna Pandohie and Dr. Damian Alexander for their invaluable assistance during the study. Their dedication and technical expertise were instrumental in the successful completion of this work.

References

1. Ali, A., and D. Villarroel-Lamb. 2020a. "A Swash-Zone Seaward Boundary Condition for Impermeable Beaches." In Sustainable Built Environment – Proceedings of the International Conference on Emerging Trends in Engineering & Technology (IConETech-2020), 1- 5 June, 2020, edited by Boppana V. Chowdary, 134- 143. St. Augustine: Faculty of Engineering, The University of the West Indies. https://doi.org/10.47412/oedt3765.
2. Amsden, A. Anthony, and Francis H. Harlow. 1970. "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows." Technical Report No. LA-4370, New Mexico. Los Alamos: Los Alamos National Laboratory.
3. Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.

ISSN: 3050-9696 E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

4. Briganti, Riccardo, Alec Torres-Freyermuth, Tom E. Baldock, Maurizio Brocchini, Nicholas Dodd, Tian-Jian Hsu, Zhonglian Jiang, Yeulwoo Kim, Jose Carlos Pintado-Patiño, and Matteo Postacchini. 2016. "Advances in Numerical Modelling of Swash Zone Dynamics." Coastal Engineering 115: 26–41. https://doi.org/10.1016/j.coastaleng.2016.05.001.
5. Gao, Donghong, N. B. Morley, and V. Dhir. 2003. "Numerical Simulation of Wavy Falling Film Flow Using VOF Method." Journal of Computational Physics 192 (2): 624– 642. https://doi.org/10.1016/j.jcp.2003.07.013.
6. Gao, Donghong, and Jin Sun. 2011. "Using DEM in Particulate Flow Simulations." In Hydrodynamics - Optimizing Methods and Tools, edited by Harry Schulz, André Simões and Raquel Lobosco, 29- 50. London: IntechOpen. https://doi.org/10.5772/26681.
7. Harlow, Francis H., and J. Eddie Welch. 1965. "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface." The Physics of Fluids 8 (12): 2182-2189. https://doi.org/10.1063/1.1761178.
8. Kolla, S. K., & Mangalampalli, B. M. (2024). Edge-Based Deep Learning Systems for Point-of-Care Diagnostic Intelligence. Journal of Neonatal Surgery, 13(1), 2387-2399.
9. Hirt, Cyril W., and B. D. Nichols. 1981. "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries." Journal of Computational Physics 39(1): 201-225. https://doi.org/10.1016/00219991(81)90145-5.
10. Johnson, N. Lloyd. 1996. "The Legacy and Future of CFD at Los Alamos." In Proceedings of the 1996 Canadian CFD Conference, 3- 4 June 1996, Ottawa, Canada.
11. Kothe, Douglas Brian, Raymond C. Mjolsness, and Martin D. Torrey. 1991. "RIPPLE: A Computer Program for Incompressible Flows with Free Surfaces." Technical Report No. LA-12007-MS. New Mexico. Los Alamos: Los Alamos National Laboratory. https://doi.org/10.13140/RG.2.1.1397.5200.
12. Kranenborg, J., G. Campmans, J. V. D. Werf, A. Reniers, and S. Hulscher. 2019. "Depth-Resolving vs Depth-Averaged Modelling of Swash Zone Hydrodynamics." In Coastal Structures 2019 Conference, 30 September – 2 October, 2019, edited by Nils Goseberg and Torsten Schlurmann, Hannover, Germany: Bundesanstalt für Wasserbau. https://doi.org/10.18451/978-3-939230-64-9_083.
13. Pamisetty, V., & Amistapuram, K. Smart Decision Support Systems For Dynamic Tax Policy Optimization Using Reinforcement Learning.
14. Krüger, Timm, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. 2017. The Lattice Boltzmann Method Principles and Practice. Switzerland: Springer Cham. https://doi.org/10.1007/978-3-319-44649-3.
15. May, Sandra, Andrew Nonaka, Ann Almgren, and John Bell. 2011. "An Unsplit, Higher-Order Godunov Method Using Quadratic Reconstruction for Advection in Two Dimensions." Communications in Applied Mathematics and Computational Science 6 (1): 27–61. https://doi.org/10.2140/camcos.2011.6.27.

ISSN: 3050-9696 E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 04
RECEIVED: AUGUST 28 REVISED: SEPTEMBER 25 ACCEPTED: OCTOBER 04 PUBLISHED: OCTOBER 29

16. Mory, M., S. Abadie, S. Mauriet, and P. Lubin. 2011. "Run-up Flow of a Collapsing Bore over a Beach." European Journal of Mechanics - B/Fluids 30 (6): 565–576. https://doi.org/10.1016/j.euromechflu.2010.11.005.
17. Mangala, N. (2024). Leveraging Microsoft Fabric lakehouse as an AI-ready data platform for enterprise analytics. Journal of Information Systems Engineering and Management.
18. Peregrine, D. H., and S. M. Williams. 2001. "Swash Overtopping a Truncated Plane Beach." Journal of Fluid Mechanics 440: 391–399. https://doi.org/10.1017/s002211200100492x.
19. Pillord, James Edward, and Elbridge Gerry Puckett. 2004. "Second-order accurate volume-of-fluid algorithms for tracking material interfaces." Journal of Computational Physics 199(2): 465-502. https://doi.org/10.1016/j.jcp.2003.12.023.
20. Prosperetti, Andrea, and Tryggvason Grétar. 2007. Computational Methods for Multiphase Flow. Cambridge: Cambridge University Press.
21. Puckett, E. Gerry, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider. 1997. "A High-Order Projection Method for Tracking Fluid Interfaces in Variable Density Incompressible Flows." Journal of Computational Physics 130 (2): 269– 282. https://doi.org/10.1006/jcph.1996.5590.
22. Rudman, Murray. 1997. "Volume-Tracking Methods for Interfacial Flow Calculations." International Journal for Numerical Methods in Fluids 24 (7): 671–691. https://doi.org/10.1002/(sici)1097-0363(19970415)24:7<671::aid-fld508>3.0.co;2-9.
23. Svendsen, I. A., and P. A. Madsen. 1984. "A Turbulent Bore on a Beach." Journal of Fluid Mechanics 148: 73–96. https://doi.org/10.1017/s0022112084002251.
24. Tomé, Murilo F., A. Castelo, J. Murakami, J. A. Cuminato, R. Minghim, M. C. F. Oliveira, N. Mangiavacchi and S. McKee. 2000. "Numerical Simulation of Axisymmetric Free Surface Flows." Journal of Computational Physics 157(2): 441- 472. https://doi.org/10.1006/jcph.1999.6348.
25. Gottimukkala, V. R. R. (2021). Digital Signal Processing Challenges in Financial Messaging Systems: Case Studies in High-Volume SWIFT Flows.
26. Tomé, Murilo F, Brian Duffy, and Sean McKee. 1996. "A numerical technique for solving unsteady non-Newtonian free surface flows." Journal of Non-Newtonian Fluid Mechanics 62(1): 9- 34. https://doi.org/10.1016/0377-0257(95)01391-1.
27. Tomé, Murilo F., and Sean McKee. 1994. "GENSMAC: A Computational Marker and Cell Method for Free Surface Flows in General Domains." Journal of Computational Physics 110 (1): 171- 186. https://doi.org/10.1006/jcph.1994.1013.
28. Tu, Jiyuan, Kiao Inthavong, and Goodarz Ahmadi. 2013. Computational Fluid and Particle Dynamics in the Human Respiratory System. Dordrecht, Netherlands: Springer.
29. Yeoh, Guan Heng, and Jiyuan Tu. 2010. Computational Techniques for Multiphase Flows: Basics and Applications. Amsterdam, Netherlands: Elsevier.