

Comparison Between Heuristic and Eulerian Interface Capturing Approaches for Shallow Water Type Flow

Hossein Aghakhani, Keith Dalbey, Abani Patra, David Salac

Abstract

Determining the wet-dry boundary has historically been a challenge when solving the depth-averaged shallow-water equations (or similar granular flows), and has been the focus of much research efforts. There are two principal approaches to capture the interface for a shallow water flow. The first method, which is the most common, uses some heuristic method to reconstruct the interface based on geometrical data like the basal depth, and numerical solution such as flow depth in each cells and its surrounding neighbors. This method is usually called volume/free-surface relationship (VFR) in the literature. The other more mathematically rigorous solution is employing an interface capturing/tracking method to find the accurate location of the flow interface. There are a number of different types of method that are generally categorized into Lagrangian and Eulerian. In the present paper, these two strategies were examined, and their results compared to show the advantages and disadvantages. This study was conducted for geophysical flows, but same conclusion could be made for any other shallow water (SW) flow.

In the heuristic solution part, a threshold was assumed as the minimum of flow depth in a cell. Based on that the cells divided into wet, dry and partially wet cells. Then fluxes were adjusted for partially wet cells. To obtain more accurate result adaptive mesh refinement/unrefinement were used specially for near boundary cells. For the second type of approaches, we used level-set and phase field methods which both are Eulerian interface capturing methods. In either of the methods a transport equation must be coupled with the system of equations in order to capture the interface of the flow. The algorithm and detail of implementation are explained in related sections. The result of the study showed that Heuristic and Eulerian interface capturing approaches have their own restrictions, but considering the computational cost and related difficulties the first approach seems to be more affordable for a wide range of SW flows.

1 Introduction

Shallow water (SW) flows include a wide range of fluid flows. In SW flows that the fluid depth is much smaller than the characteristic length of the fluid body. In driving the corresponding governing equations, the shallowness of flow lets the distribution of the state variables neglected in the perpendicular direction to the basal surface, and the effect of this direction replaced with an integrated average in the equations. This mathematical simplification causes the 3D to be represented in 2D equations. The aforementioned condition holds for many circumstances in geophysical flow and the same conservation equations with some minor

modifications could be used to study these type of flows.

To this aim, Eglit and Sveshnikova [?] modified the depth-averaged Saint Venant equations to simulate granular snow avalanches, and almost a decade later Savage and Hutter[?] were the first to adapt the depth averaged Saint Venant equations to accommodate geophysical mass flow.

Since this type of flow is free moving boundary flows, one of the basic difficulties of the numerical solution of them, which itself causes a large group of other difficulties, is finding the location of flow interface. More clearly, the governing equations are just valid in wet areas, so there must be a way to discriminate between wet and dry areas in numerical simulation. This problem is usually called the **wetting and drying (WD)** or **Thin-Layer** problem in SWE context, and the lack of a solution for this problem leads to the formation of a non-physical super shallow flow layer in the domain. This non-physical thin layer could extend large distances from the realistic main body of the flow, which can cause inaccurate construction of the boundary or sever numerical instabilities in the numerical solution.

Beside the numerical issues, determining probable flow extents through numerical simulation is both possible and critical for many SW problems, especially for geophysical flow. For example in preparing a hazard map for a volcano or a flood it is crucial to know, where is the exact location of the front of the flow? Does the flow reach to the specific location? What is the distance of high risk locations from the infrastructures?

The answer of all above questions is not possible without good information about the interface of the flow during its traveling path.

The following problems are the major difficulties that arise in numerical solution of SWE, and most of them are made or related to the thin layer problem:

1. Numerical spreading of flow faster than its physical speed.
2. Non-physically thin layer (orders of magnitude thinner than a grain of sand). This result from the combination of conservation of mass and the numerical wicking away of material.
3. Non-physically high flow speeds. Since the solution of SWE are momentums, $\{hV_x, hV_y\}$, to find velocities, $\{V_x, V_y\}$, the momentums must be divided to flow depth, h , which causes a numerical error in an already small h , and result in overly large velocities.
4. Loss of numerical stability. Due to the above reasons, wave-speeds can become infinite at the flow boundary which means that flow equations lose their hyperbolicity character at the vacuum state interface.

A simple review on the related literature showed that numerous attempts were made to overcome this problem over time especially in the recent years. Here we will briefly review the topic and refer the interested reader to [?, ?] for more information.

Generally, all of the implemented method for the issue could be categorized in two basic groups of what we call here heuristic methods and interface tracking/capturing methods. The basic idea in the first group is to reconstruct the flow interface based upon some simple heuristic or some algebraic relations which are not

essentially related to the physics of the problem. In interface tracking/capturing approaches, an auxiliary equation/equations is coupled with the original system of equations in Eulerian or Lagrangian frame of view, to follow the flow interface through its path over time. The reason that we mention here both tracking and capturing is that they are usually used for Lagrangian and Eulerian methods respectively. Each of these major groups have their own subcategories which could be found in [?, ?, ?, ?] references, but they will be explained briefly here to provide some insight that is needed in the next parts.

All of the methods that we call here heuristic methods can be divided in one of following four strategies [?]:

1. Filling the entire computational domain with a thin layer of fluid
2. Using a depth scale to check whether a cell or a node is wet or dry (or possibly partially wet), and then making a decision to add or remove it from the computational domain.
3. Employing some extrapolating scheme from the wet cells into their neighbor cells to approximate the location of the interface. This method is usually called volume/free-surface relationship (VFR) in the literature.
4. Permitting fluid height to be negative, which means that it is below the topographical surface.

Each of the above strategies might be more useful for its specific case. Table 1 compares them very briefly.

Strategy	Mass Conservation	Physics
Thin film	Adequate, but requires correction after water levels obtaining	produce a smooth and realistic wetting front
Cell removal	Dependent on numerical method for solving the equations	Excellent, performs better on advancing front than receding front
VFR	Conservative, with aid of some correction procedure	Very good in wide verity of problems
Allowable negative depth	Conservative, but performance depend on WD parameters	Same as mass conservation

On the other hand, with the interface tracking/capturing methods one can find the interface of a moving boundary flow in a more rigorous way. Based upon the employed frame of work, these methods are divided into Lagrangian and Eulerian methods. Roughly speaking in Lagrangian type methods, the initial interface is replaced with some of the material points (particles) that represens the interface. These points move with a velocity vector field which comes from numerical solution at each time step. The interface is constructed at each time step by connecting the updated location of these points. Mark And Cell method (MAC)[], Simplified Marc And Cell method (SMAC), and Surface Marker method are some examples of Lagrangian interface tracking methods. Obviously, the accuracy of the method is highly depend on the number of particles that form the initial boundary. High computational cost, tendency to instability and inability to track the complex topological changes are the most important drawbacks of these methods. However, these methods are more useful as an auxiliary tool to compensate the defects of Eulerian methods such as conservation of mass

in level set method. Tai et el. [?] is an example of the application of this method for 1D Savage-Hutter equation. They used a non-oscillatory central (NOC) difference scheme with a stationary uniform mesh, and achieve good results by augmenting this solver with a Lagrangian front tracking scheme.

The other important type of interface capturing schemes are Eulerian methods. Level set, phase field and VOF are the most well-known methods in Eulerian frame of work. In these types of methods the interface is captured by solving another transport equation for a scalar that explicitly, in level set method, or implicitly, in VOF and phase field method, indicates the interface location. This order parameter is coupled with the other governing equations of primitive variables to jointly present a rigorous model for the interested problem.

Hirt and Nichols[?] were the first to propose a VOF method. In this method, the mentioned scalar variable is the fraction/volume of a particular fluid in each cell. To construct the interface based on this method, the interface surface reconstruction technique shall be performed at each time step. Youngs[?] achieved a significant improvement in VOF by adding a piecewise linear interface calculation (PLIC) representation of the fluid boundary. Youngs' method has been shown to be robust and efficient, but it is only first-order accurate. More advanced methods about this issue are available in [?] and Gopala and van Wachem[?] [?]. Level set method was introduced by Osher and Sethian in 1988. In this method the scalar variable is a signed distance function which shows the distance of each point to the interface. This distance is defined such that is positive for outside of the boundary points and negative for inside of the boundary points. The solution of this equation will show the place of interface at each time step, but since the solution of the original equation is not essentially a signed distance function another equation has to be solved to make the obtained result a signed distance function. This technique is called initialization and reinitialization technique. There are also a number of other methods like fast marching or combination with Lagrangian schemes to get a more accurate result. The main issue about this technique is that the basic method is not mass conservative and some other complementary methods have to be used to preserve the mass or volume in numerical solution.

Quecedo and Pastor[?] used level set method in their work to fix the WD problem for a SW problem. They developed Taylor-Galerkin approach to solve the SWEs. They presented two different options for handling the wetting-drying areas. The first approach used was described as a “simple yet efficient” method which falls into heuristic method type approaches. Although they claimed that they used level set as the second approach, they did not report any result of that. They finally concluded that the level set is expensive and unnecessary for their problem.

The last method that we will overview here is phase filed method. Phase field is another Eulerian type interface capturing scheme that the interface is implicitly related to the order parameter which shows the phase variation on the domain. Usually, this value varies between 0 to 1 or -1 to 1, in such a way that it is 1 if the cell is wet, it is zero (or -1) if the cell is dry, and it is something between 0 to 1 (or -1 to 1) for partially wet cells. This method is more developed and popular in material science research, and drew the attention of the researchers in many disciplines especially in the recent years. Cahn-Hilliard and Allen-Cahn formulations

are two principal formulation for this method. The basic difference between these two form is that the first one is mass conservative, and the second one is not. Mathematically, the conservative form has a forth order derivative while the other one has a second order derivative. This difference in the highest order of derivative will make the laterform much easier to implement. As a result people first try to use the second one and then if they did not succeed, they will proceed to the other one. In this method the interface is a diffusive region between the phases in spite of level set which yields to a sharp interface. Phase field equation is derived from the relation between the changes of the phases and free energy level of the system. The strong physics behind that makes it powerful particularly in simulating energy and thermodynamics related problems. Based on our best knowledge no report was published earlier that uses this method for capturing the interface in SWE.

In this work we used some combination of the first group of approach and then compared these results with the results of the phase field and the level set methods. We tried to stick to the basic methods, as long as the quality of the result was reasonable and it does not increase the computational cost significantly. The basic procedure in the heuristic approach is: First, use a very small threshold to discriminate between wet and dry cells, then divide the cells into three groups of wet, dry and partially wet cells. The next step is to adjust the fluxes for the partially wet cells and the last step is to update the state variables for wet cells.

In the level set part, the basic level set was implemented [1] with the reinitialization scheme introduced by [2] for every 20 time steps. To solve both of these hyperbolic equations we used a forst order accurate solver.

For the phase field method, we chose Allen-Cahn formulation. The expensive part of the Allen-Cahn is the laplacian term that makes it non affordable for explicit time integration schemes. To handle this difficulty, operator splitting technique was used, which allows one half to be solved by an implicit scheme, and the rest of the terms were updated in the default explicit integrator.

As a general comparison between the methods it could be mentioned that the first approaches are less expensive, but since they do not have a rigorous physical background, the obtained interface not be realistically pleasurable. On other hand, the capturing interface methods have stronger mathematical and physical structure, and they can be coupled with the other equations to be employed for a more complicated case. However, these types of approaches are more computationally expensive, and are harder to implement and couple with the basic solver and the other parts of the code.

The structure of the rest of this paper is as follows: in next part the SWEs for the geophysical flows and the basic features of the solver are introduced. In section 3, different methods that were employed to mitigate WD problem for SWE are explained. After that the obtained results from the methods are compared, and finally the conclusion of the work is provided.

2 Governing Equations

The Savage-Hutter equation for geophysical flows was first introduced in the late 1980's. Their original model has subsequently been improved upon by Savage_Hutter themselves as well as others [?, ?, ?, ?, ?, ?]. In our earlier work [?, ?, ?], we developed the Titan2D depth-averaged geophysical flow simulator. Titan2D is a parallel with dynamic repartitioning, high order, slope-limiting, upwinding, two-dimensional Godunov solver (without splitting), with adaptive mesh refinement and Geographic Information System (GIS) integration which lets it use Digital Elevation Models (DEMs) of real terrain. While our new multi-faceted thin-layer mitigation strategies were developed in the context of Titan2D's capabilities, much of this approach should be appropriate for use in depth-averaged flow solvers with different numerical implementations.

The depth-averaged equations that Titan2D solves are:

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial(V_x \cdot h)}{\partial x} + \frac{\partial(V_y \cdot h)}{\partial y} &= S_h \\ \frac{\partial h V_x}{\partial t} + \frac{\partial(V_x \cdot h V_x + 0.5 k_{ap} g_z h^2)}{\partial x} + \frac{\partial(V_y \cdot h V_x)}{\partial y} &= S_x \\ \frac{\partial h V_y}{\partial t} + \frac{\partial(V_x \cdot h V_y)}{\partial x} + \frac{\partial(V_y \cdot h V_y + 0.5 k_{ap} g_z h^2)}{\partial y} &= S_y \end{aligned} \quad (1)$$

In these equations:

- The coordinate system (see Figure 1) is aligned such that x and y are tangential directions to the surface of the 3D terrain and z is normal to the surface.
- The effect of terrain elevation is represented by gravitational source terms.
- h is the flow depth in the z direction that hug the terrain.
- hV_x and hV_y are the components of “momentum” respectively in x and y directions.
- k_{ap} is a highly nonlinear term that shows the effect of the earth pressure coefficient, which has active (diverging $\mp = -$), passive (converging $\mp = +$), and neutral (neither diverging, nor converging, $k_{ap} = 1$) conditions.

$$k_{ap} = 2 \frac{1 \mp \sqrt{1 - \cos^2(\phi_{int}) (1 + \tan^2(\phi_{bed}))}}{\cos^2(\phi_{int})} - 1 \quad (2)$$

Note that in the momentum equations $k_{ap} g_z \frac{h}{2} h$ is the contribution of hydrostatic pressure to the momentum fluxes. S_h is a source of mass, i.e. material that either effuses or erodes out of the ground. S_x is the sum of a gravitational driving force, the friction that resists motion of the material relative to the bed, and the

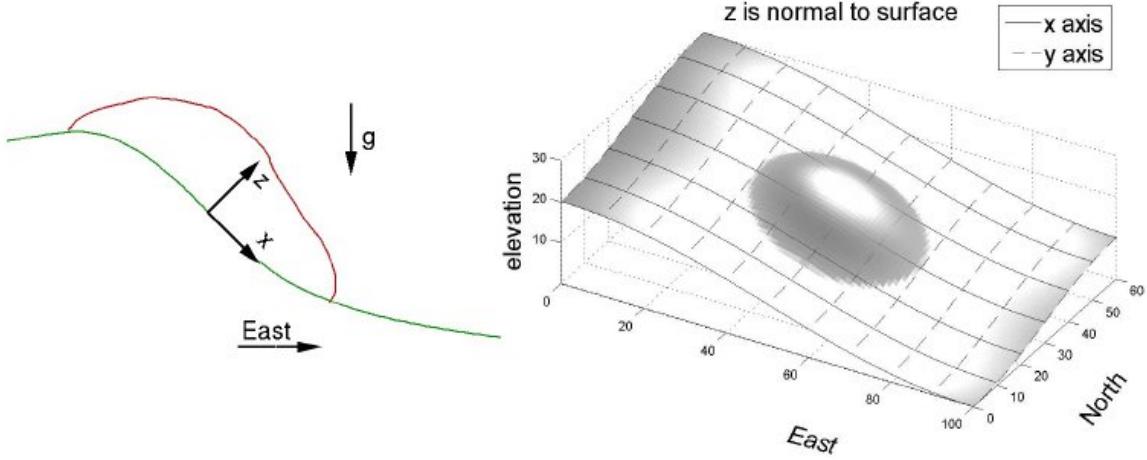


Figure 1: In the local coordinate system the z direction is normal to the surface; the x and y directions are tangential to the surface.

friction that resists the internal shearing motion of the material see eq. (3).

$$\begin{aligned} S_x &= g_x h - \frac{V_x}{\sqrt{V_x^2 + V_y^2}} \max \left(g_z + \frac{V_x^2}{r_x}, 0 \right) h \tan(\phi_{bed}) - \operatorname{sgn} \left(\frac{\partial V_x}{\partial y} \right) h k_{ap} \frac{\partial(g_z h)}{\partial y} \sin(\phi_{int}) \\ S_y &= g_y h - \frac{V_y}{\sqrt{V_x^2 + V_y^2}} \max \left(g_z + \frac{V_y^2}{r_y}, 0 \right) h \tan(\phi_{bed}) - \operatorname{sgn} \left(\frac{\partial V_y}{\partial x} \right) h k_{ap} \frac{\partial(g_z h)}{\partial x} \sin(\phi_{int}) \end{aligned} \quad (3)$$

Titan2D solves the above system of equations with a finite volume Godunov method that is demonstrated in eq. (4).

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \{ F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n \} - \frac{\Delta t}{\Delta y} \{ G_{i+\frac{1}{2}}^n - G_{i-\frac{1}{2}}^n \} \quad (4)$$

In the above equation, G and F are the flux terms at the inter-cell boundaries which are computed by HLL Riemann solver that is explained in the next section.

2.1 Using Appropriate Riemann Fluxes

For flux computation TITAN2D uses HLL solver. HLL solver is an approximate Riemann solver that just use the maximum and minimum of characteristic speeds of a hyperbolic system to build the conservative fluxes at the interface.

Following the recommendation of Toro[?], Titan2D uses the HLL Riemann average of fluxes. Let U be a state variable, F be a flux of that state variable, s be a maximum wave speed, and R and L subscripts which

denote “right” and “left” values respectively. The HLL Riemann flux is then

$$F_{HLL} = \begin{cases} F_L & \text{if } s_L \geq 0 \\ F_R & \text{if } s_R \leq 0 \\ \frac{s_R F_L - s_L F_R + s_L s_R (U_R - U_L)}{s_R - s_L} & \text{otherwise} \end{cases} \quad (5)$$

For Savage-Hutter system of equation the characteristic speeds are: $s_{1,3} = v \pm a$ and $s_2 = v$ where v is the velocity of fluid in the corresponding direction and $a = \sqrt{k_{apgh}}$. But just the above HLL solver is not enough for solving this system of equations. Fraccarollo and Toro[?] note that at the wet-dry boundary the Riemann solution consists of a single rarefaction wave whose speed (when averaged over the cell length) is bounded above by the cell average flow speed plus **twice** the “speed of sound,” $s = v + 2a$, where $a = \sqrt{k_{apgh}}$ for shallow water type granular flows. They also note that “an overestimate of the true wave speeds results in enhanced stability” while an “underestimate of the true wave speeds could be fatal” to stability. They became the first to solve this problem by constructing an approximate Riemann solver.

2.2 Adaptively Meshing

In the Titan2D, each grid cell is square or very nearly so. Rather than using a uniform mesh, different sizes of cells are allowed, with each successive “generation” covering one fourth the area of its “parent” cell. Note that only one generation of irregularity is allowed between a cell and its neighbors.

At the beginning of the simulation, i.e. before the first update, the boundaries of all piles are maximally refined. An initial mesh for a simulation at Colima Volcano, Mexico, is displayed in Figure ??.

During normal adaptivity, cells are selected for refinement if they meet either of two requirements:

1. They have large, as compared to the average, inter-cellular fluxes.
2. They are at or within a few cells of any of four small, non-dimensional flow depths (multiples of `GEOFLOW_TINY`).

The former condition allows for accurate capture of sharp changes in state variables. The latter results in banded regions of the flow separated into “rings” of maximally refined buffer cells. This limits artificially high transportation of material from the larger depth bands to the lower depth bands. The outermost ring is at a flow depth of `GEOFLOW_TINY`. The rings are a number of cells wide equal to the number of iterations between mesh adaptations. This guarantees that material will only flow into maximally refined dry cells at the wet-dry front. However, material with flow depth less than `GEOFLOW_TINY` may be left outside the outermost band of buffer cells by the passing flow. The rings of the later buffer cells strategy decreases the numerical wicking (Problem 1 from Subsection ??) both within the flow and at the boundary. When combined with thresholding, it is sufficient to decrease Problems 1, 2, and 3 to a level that prevents the loss of stability due to Problem ??.

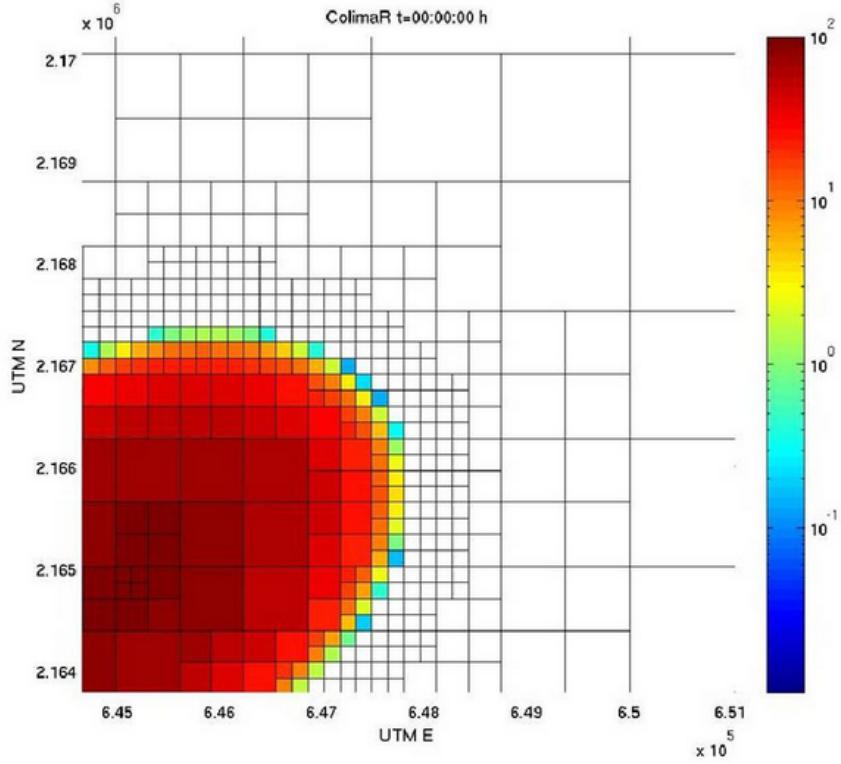


Figure 2: Buffer cells

Unrefinement takes place immediately after refinement; a group of four “brother” cells will be selected to merge into their mutual “father” cell if the sum of their inter-cellular fluxes is very low compared to the average flux. Cells that have either just been refined or are otherwise in any of the current bands of buffer cells are immune to unrefinement.

3 Solution

In the aforementioned parts the major difficulties of numerical analysis of the shallow water equations were discussed, and it was shown that most of them are directly or indirectly related to the WD problem. For more intuition, the result of a simulation for Atenquique debris is displayed in figure 3. The left picture is the obtained result form the solver and the right picture is the same result, but flow height is displayed only for flow height greater than 0.5 m. This picture shows for the pure solver, that a dominant part of the domain is filled with a negligible flow depth and this makes the actual interface mix with the thin layer region, and also causes some other problems. In this section, three implemented methods to solve the WD problem which finally yields to mitigate the numerical difficulties of SWE are explained.

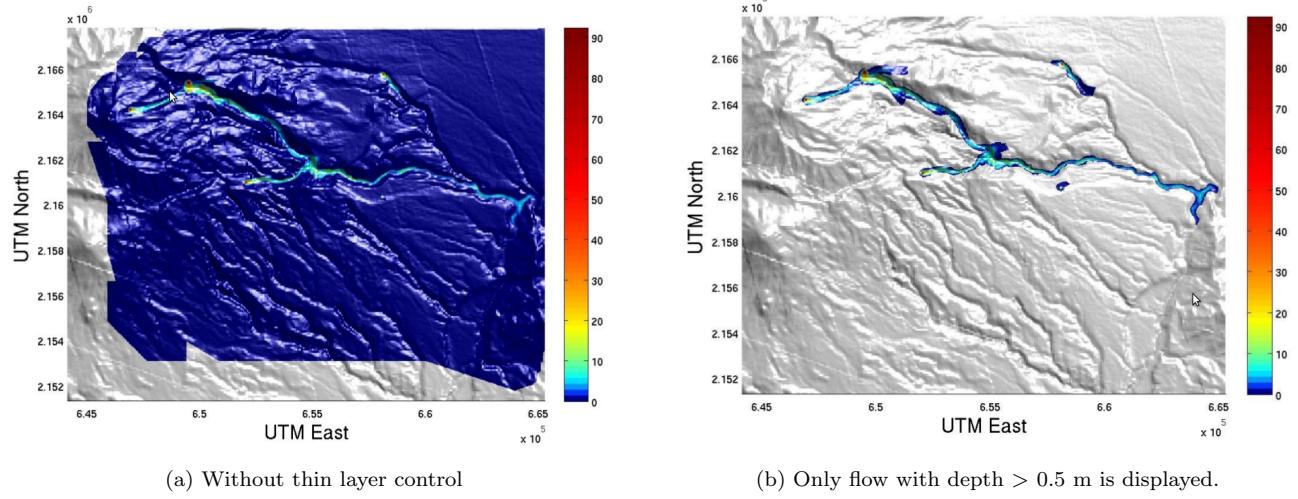


Figure 3: Thin layer problem in maximum over time flow depth simulation of the 1955 debris flow at Atenquique, Mexico.

3.1 First: Heuristic method

In this part of the paper the first method that has been used to solve the WD problem in the SWE is explained.

3.1.1 Selecting appropriate Threshold

In this approach we use a very small threshold to segregate the wet, partially wet and dry cells. The critical contribution of this scaling is to allow other facets to identify where the flow is non-physically thin, i.e. where should they consider the boundary of the flow to be. Consistency is the key requirement of whatever strategy we choose to determine the scaling. That is, the chosen strategy must be able to generate the same “appropriate” value for depth scale at the beginning and at the end of a simulation.

In a “typical” geological simulation, say of the collapse of a volcanic dome which would be modeled in Titan2D as a “pile source,” the initial body of mass can be quite deep, but at the end of the simulation the material will likely be spread over a large area. Hypothetically, if one were to scale by maximum flow depth at the beginning and end of the simulation, they would obtain vastly different values. More importantly, if the only source of mass was an effusion of material out of the ground, the maximum initial flow depth would be zero, even if the total volume during the course of the simulation was the same as the previously mentioned “pile source.”

On the other hand, scaling by the cube root of the total volume of the flow being simulated is entirely consistent and is the flow depth scaling factor used by Titan2D. We do **not** claim that the cube root of

volume is any more or less appropriate as a scaling factor than maximum initial flow depth in other shallow water contexts, for example storm surge simulation.

Having chosen a consistent scaling factor, we were then able to define associated non-dimensional depths for negligibly thin and merely thin flow. If one were to assume that a particular geophysical mass flow event involved a volume of $10^8[\text{m}^3]$, it would then be reasonable to state that flow depths of less than $5[\text{cm}]$ were both negligible and non-threatening. This roughly equates to non-dimensional negligible flow depth, which we call **GEOFLOW_TINY**, with the value

$$\text{GEOFLOW_TINY} = 0.0001 \quad (6)$$

Using this value, and assuming the volume used in a laboratory scale test was $1[\text{cm}^3]$, the resultant negligible flow depth would be $0.0001[\text{cm}]$. As can be observed from these two examples, the chosen value of **GEOFLOW_TINY** is physically appropriate across a very large range of volumes. We therefore use a theoretical contour at this depth as the boundary of the simulated flow.

In introduction section we stated that, unless steps are taken to prevent it, the numerical wicking (Problem 1) will cause the flow to spread 1 cell every time-step even though the product of flow speed and time-step size is less than the cell length. In addition, the familiar continuum equations loose hyperbolicity (wave speeds tend to infinity) at the boundary between a continuous material and a vacuum. Having implicitly defined the flow boundary, we prevent calculation of state variables outside the flow, i.e. in regions with flow depth below **GEOFLOW_TINY**. This reduces both non-physical flow spreading and computational cost. Note that state variables in cells with flow depth less than **GEOFLOW_TINY** are **not** zeroed.

3.1.2 Interface Reconstruction

As noted above, the use of a standard Eulerian grid imposes discrete fixed increments to flow extent, which contributes to the wicking problem (Problem 1) at the boundary. Use of adaptive mesh refinement reduces this error by reducing the size of the increment where it is most beneficial to do so. The Lagrangian approach does not have this limitation, which Tai et al [?] took advantage of when they augmented their one dimensional NOC scheme with Lagrangian front tracking. However, implementing a hybrid Eulerian-Lagrangian scheme in two dimensions is significantly more complex.

Therefore as part of our multi-faceted approach, we implemented a very simple and inexpensive interface reconstruction and predictive Lagrangian front tracking scheme. Knowledge of the interface allows us to generate more a representative average, for an individual cell edge over the time-step, and for values of state variables which are then used to compute inter-cellular fluxes into/out-of partially wet cells. The interface reconstruction scheme is illustrated in figure 4.

Specifically, each partially wet cell is assumed to be split by a straight line into a completely dry and a

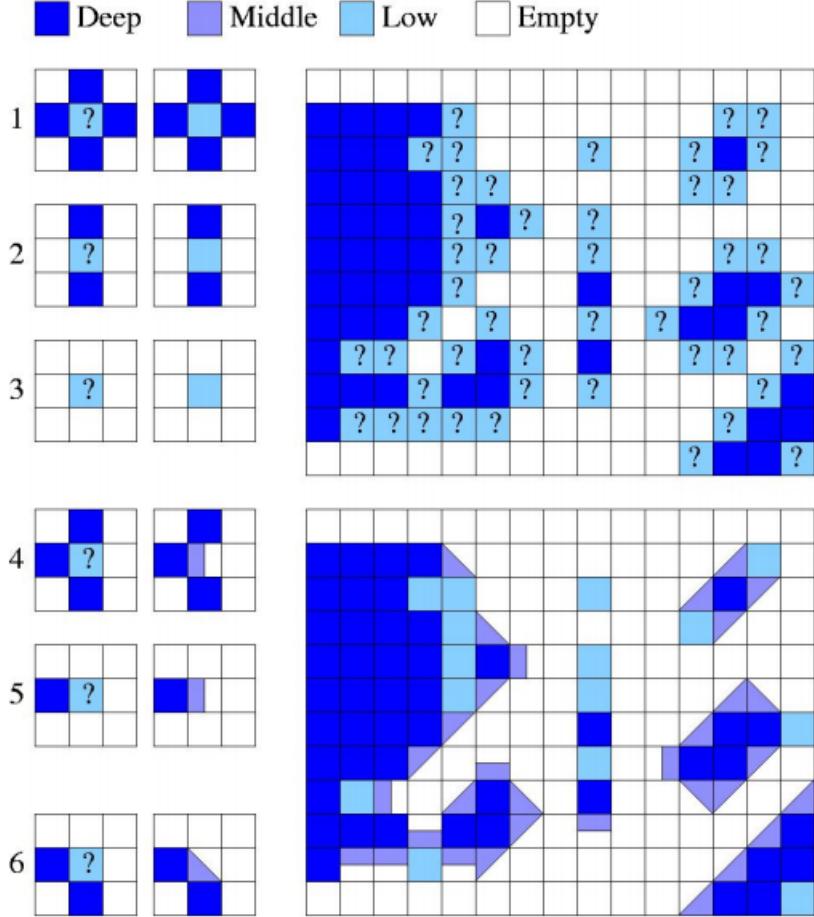


Figure 4: Interface Reconstruction in TITAN2D

completely wet part. For the sake of simplicity, we restrict this line to one of four orientations: east-west, north-south, or parallel to either diagonal of the square cell. However, all placements/translations of the wet-dry line are allowed. At the beginning of each time-step, the orientation of the line for the entire time-step is set based solely on which of the cell’s neighbors have flow depth greater/less than the `GEOFLOW_TINY` threshold¹. Orientation of the wet-dry line, and which side of it is wet, is indicated by a single integer representing the geometrically determined “most wet node.” The only nine possible values for the most wet node are be any of the cell’s corners, edge midpoints or its center. The most wet node is assumed to have a flow depth that is the maximum of the cell or any of its four neighbors. The placement of the line is such that the volume of material in the cell under a plane passing through the most wet node and the wet dry-line is the same as the unadjusted cell.

¹As stated in Subsection 2.2, the refinement strategy ensures that the flow front will always be maximally refined. This simplifies the coding since all cells at the front can be safely assumed to have only one neighbor on each side.

Since we now know where the wet-dry line is at the beginning of the time step, we can compute which of the cell's edges are completely wet, completely dry, or partially wet, and the fraction of wetness for the partially wet edges. Given the orientation and beginning of time-step location of the wet-dry line and the cell's state variables, it is also fairly straight forward to predict the line's end of time-step location. This is done by convecting the wet-dry line at the shock speed, $s = v + a$ where the speed of sound has been adjusted to account for only part of the cell being wet, i.e. $a = \sqrt{k_{ap}gh\frac{A}{A_{wet}}}$. Here A is the whole cell's area and A_{wet} is the portion of the cell's area that is wet.

Since the wet-dry boundary within each partially wet cell is assumed to be a straight line width, during the time-step, fixed orientation, convecting a single point, in this case the midpoint, on the line is equivalent to convecting the entire wet-dry line. A spatial and time average of the wetness factor for the edge over the time-step can therefore be easily computed as

$$W = \left(\frac{0 \cdot \Delta t_{dry} + \frac{1}{2}(w_{beg} + w_{end})\Delta t_{part} + 1 \cdot \Delta t_{wet}}{\Delta t} \right) \left(\frac{A}{A_{wet}} \right) \quad (7)$$

Where $\Delta t = \Delta t_{dry} + \Delta t_{part} + \Delta t_{wet}$ is the entire time-step, Δt_{dry} is the portion of the time-step for which the edge is completely dry, Δt_{part} is the portion of the time-step for which the edge is partially wet and partially dry, Δt_{wet} is the portion of the time-step for which the edge is completely wet, w_{beg} is the edge's fraction of wetness at the beginning of the time-step, and w_{end} is the edge's fraction of wetness at the end of the time-step.

3.1.3 Adjusting Fluxes in Partially Wet Cells

The state variables used to compute the physical fluxes are the whole cell average values multiplied by the edge wetness factor. Note this results in the zeroing of this cell's physical fluxes for its sides that will be completely dry for the entire time-step and increasing them for sides that will be completely wet for the entire time-step. The numerical fluxes are then taken to be the HLL Riemann average of the “adjusted” physical fluxes from the cells on both sides the edge.

The wetness factor adjustment of fluxes delays/decreases the transfer of material from partially wet cells to completely dry cells. This delay significantly reduces the amount of non-physical flow spreading, and requires negligible additional computation and memory usage. In terms of memory usage, this scheme only requires one additional integer indicating the most wet node, and two additional decimal numbers for the cell's fraction of wet area and the location of the wet-dry line's midpoint (implemented as a single number ranging from zero to one). The flux adjustment in partially wet cells mitigates the numerical wicking (Problem 1) at the boundary but not within the flow (that requires either a uniform grid or an adaptive strategy based on fluxes and the rings of buffer cells strategy).

3.2 Second: Phase Field Method

As noted earlier, the other approach that is employed for capturing the interface of a SW type flow is phase field method. In this method, a continuous order parameter is augmented into the state variables. This order parameter that is displayed here with φ , implicitly represent the interface in the domain. To this aim, a new transfer equation must be solved which is coupled with the other state variables. Papers [?, ?] are good references about the history and evolution of the method. Phase diffusion methods and particularly phase field method is base upon the notion that the interface between phases is a diffusive region rather than a sharp interface with a finite width. As described in the introduction 1 of this paper, ϕ is constant inside the bulk phases but changes smoothly between the phases. In this work ϕ is 1 for the fluid phase and it is 0 for void regions and is between 0 to 1 on the diffusion region. The strong thermodynamical and physical derivation of the method make it so powerful for capturing the complicated topological changes especially for the problems that the interface motion depends on gradients of an external field normal to the interface and on the local curvature of the interface [?]. Cahn-Hilliard and Allen-Cahn are the two principal phase field formulations that are displayed in eq. (8) and eq. (10) respectively. The derivation of the equations in more detail could be found in [?, ?]. As could be seen in these equations, the time material derivative of phi is related to the product of the mobility and derivative of the free energy with respect to ϕ . The basic difference between these formulations is that the Cahn-Hilliard is mass conservative but the Allen-Cahn is not. On other hand, there is a biharmonic term in the right hand side of the Cahn-Hilliard equation which is forth order derivative and makes it more difficult to implement. Usually, people try to use the Allen-Cahn first and then if the result was not satisfactory they switch to the Cahn-Hilliard phase field. Since in our work the computational cost has a higher priority than an accurate solution for the interface the Allen-Cahn form was chosen.

$$\frac{\partial \varphi}{\partial t} + \vec{V} \cdot \nabla \varphi = \nabla \cdot (M(\varphi) \nabla \mu(\varphi)) \quad (8)$$

which

$$\mu = F'(\varphi) - \varepsilon^2 \Delta(\varphi) \quad (9)$$

In the above equation F is Helmholtz free energy of a unit volume of homogeneous material of composition φ .

$$\frac{\partial \varphi}{\partial t} + \vec{V} \cdot \nabla \varphi = \frac{1}{Pe} (-f(\varphi) + \varepsilon^2 \Delta \varphi + \xi(t)) \quad (10)$$

As discussed before, the Allen-Cahn formulation is not mass conservative, so to conserve the mass a Lagrange multiplier is added into the equation as a source term that is displayed in eq. (10) by $\xi(t)$
Depending on the problem conditions there are different choices for the mobility function, such as $M(\varphi) = 1 - \varphi^2$ or $M(\varphi) = 1$ []. For this work we selected the later one.

The default time integration scheme in TITAN2D is forward Eulerian scheme. This scheme is less expensive and fast but conditionally stable. Satisfaction of CFL condition is the necessary condition for stability in this

scheme. Considering that the phase field method has a second order derivative (laplacian term), to satisfy CFL condition which in this case is related to the inverse square of the spatial increment, $CFL \propto \Delta x^{-2}$, time increment must be proportional to the square of the spatial increment, $\Delta t \propto (\Delta x)^2$. Which makes the time step too small and causes an affordable simulation time. Thus the time integration scheme must be changed to another time scheme which allows larger time steps. The Eulerian implicit time scheme is unconditionally stable, but the disadvantage of this method is a system of linear equations has to be solved at each time step which impose a high computational cost. To overcome this problem we used operator splitting method to either make the time scheme stable for larger time steps, and maintain the computational cost low. By the operator splitting scheme that we used here the time integration of laplacian term is performed with Eulerian implicit scheme and the rest of terms are being updated in explicit time integrator. This solution let us to select larger time steps for implicit solver, let say for example quadruple of explicit time step, and consequently the system of equations has not to be solved at each time step. However because the solver needs to know for what cells the state variables have to be updated, the solution of the phase field is required at each time step. Therefore an extrapolation (or prediction step) based on the Taylor series was computed for intermediate time steps to approximate the phase field solution between the implicit solver time steps. This is basically a predictor-corrector scheme which is used broadly for advection-diffusion equations solvers. The formulation of the implicit and explicit solver is shown in eq. (??) and (15), and figure ?? shows the operator splitting time integration schematically.

$$\begin{aligned} \frac{\partial \varphi_1}{\partial t} + \vec{V} \cdot \nabla \varphi_1 &= f(\varphi_1) \\ L_1 = V_x \frac{\partial}{\partial x} + V_y \frac{\partial}{\partial y} \\ \frac{\varphi_1^{n+1} - \varphi_1^n}{\nabla t} &= f(\varphi_1^n) - L_1 \varphi_1^n \\ \varphi_1^{n+1} &= \nabla t(f(\varphi_1^n) - L_1 \varphi_1^n) \end{aligned} \tag{11}$$

$$\begin{aligned} \frac{\partial \varphi_2}{\partial t} &= \alpha \nabla^2 \varphi_2 \\ L_2 = \alpha \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \\ \frac{\varphi_2^{n+1} - \varphi_2^n}{\nabla t} &= \alpha L_2 \varphi_2^{n+1} \\ (1 - \nabla t \alpha L_2) \varphi_2^{n+1} &= \varphi_2^n \end{aligned} \tag{12}$$

To preserve the scalability of TITAN2D, for solving the equations we used PETSc [] library. GMRES solver which is a iterative Krylov subspace solver was exploited to solve the system of equations. To decrease the computational cost even more, we used matrix-free method to compute the laplacian term. Since the Krylov subspace solvers just need the result of matrix-vector multiplication instead of an explicit definition of matrix of coefficients, matrix-free method can be used to get the result of this multiplication by doing some algebraic operation on the participant vector in the production. This method leads to a significant save in required memory cost.

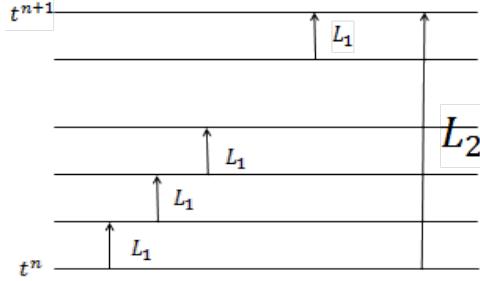


Figure 5: This picture displays the schematic time integration scheme

3.3 Third: Level Set Method

The last test method for tracking the interface for a SW flow is the Level set method. Level set method is another Eulerian interface capturing method that was introduced by Osher and Sethian in 1988 [1]. The basic of the method is to capture the interface by means of solving a hyperbolic Hamilton-Jacobi PDE on the computational domain which follows the evaluating of the boundaries. The level set variable $\Psi(X, t)$ explicitly represents the interface. In this method, $\Psi(X, t)$ is defined such that its value is zero on the interface, and changes in the domain with respect to the distance of each point to the zero level set or interface of the flow. In other words, $\Psi(X, t)$ is a distance function, so that its value is negative for the points inside the boundaries, positive for points outside of the boundary, and is zero on the interface. This method is not originally mass conservative, but with doing some techniques like fast reinitialization mass conservation could be achieved [2]. To derive the level set equation, given the initial location of the boundary an initial signed distance function on the domain which could be found by the initialization techniques that will be discussed later. The evolution of initial boundary in a moving continuum Eulerian frame of work only depends on the normal velocity of the boundary F , and the tangential velocity does not cause any topological change. Consequently, the following equation could be written for the $\Psi(X, t)$:

$$\frac{\partial \Psi}{\partial t} + F |\nabla \Psi| = 0 \quad (13)$$

Substituting $F = \vec{V} \cdot \vec{n}$, and $\vec{n} = \frac{\nabla \Psi}{|\nabla \Psi|}$ relations into eq. (13) leads to (14)

$$\frac{\partial \Psi}{\partial t} + \vec{V} \cdot \nabla \Psi = 0 \quad (14)$$

Equation (14) is a non-conservative hyperbolic equation. We used a first order upwind scheme that is explained in detail in [3] to solve this equation.

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \{F_{i+\frac{1}{2}}^- + F_{i-\frac{1}{2}}^+\} - \frac{\Delta t}{\Delta y} \{G_{i+\frac{1}{2}}^- + G_{i-\frac{1}{2}}^+\} \quad (15)$$

Where F and G are the flux terms and are coming from the flux term for in upwind scheme:

3.3.1 Reinitialization

To generate the initial signed distance function in the beginning and also to keep the solution of the level set equation during the time, we need a procedure that is called initialization or reinitialization. There are several techniques to make the obtained solution a signed distance function [?], but what we implemented here is the method presented by Sussman et al [?]. In this reinitialization technique another hyperbolic equation is solved to adjust the Ψ value in the domain:

$$\frac{\partial \Psi}{\partial \tau} + sign(\Psi)(|\nabla \Psi| - 1) = 0 \quad (16)$$

In eq. (16) τ is a pseudo-time and the equation should be solved until it converges reasonably. As it is clear from the equation the result of this equation does not change the zero level set, but it adjusts the other level sets in a sense that $|\nabla \Psi| = 1$. The method that introduced by ?? was used to obtain the solution of reinitialization equation. Since the reinitialization is not required for every time step, the reinitialization was performed after every five time steps.

$$\Psi_{ijk}^{n+1} = \Psi_{ijk}^n - \Delta t \left(max(F, 0) \nabla_{ijk}^+ + min(F, 0) \nabla_{ijk}^- \right), \quad (17)$$

where

$$\begin{aligned} \nabla_{ijk}^+ &= [max(D^{-x}\Psi_{ijk}^n)^2 + min(D^{+x}\Psi_{ijk}^n)^2 \\ &\quad max(D^{-y}\Psi_{ijk}^n)^2 + min(D^{+y}\Psi_{ijk}^n)^2]^{1/2} \end{aligned} \quad (18)$$

$$\begin{aligned} \nabla_{ijk}^- &= [min(D^{-x}\Psi_{ijk}^n)^2 + max(D^{+x}\Psi_{ijk}^n)^2 \\ &\quad min(D^{-y}\Psi_{ijk}^n)^2 + max(D^{+y}\Psi_{ijk}^n)^2]^{1/2} \end{aligned} \quad (19)$$

In the above equations D^+ , and D^- are respectively the backward and forward differences in the corresponding directions.

4 Results

4.1 inclined plane

In this section the results of the study is presented in the same order they were introduced in the paper. Table 1 shows the applied initial condition for these results.

Maximum pile height	.06 m
Major extent of the pile	.06 m
Minor extent of the pile	.06 m
Bed friction angle	32.47°
Initial friction angle	37.3°

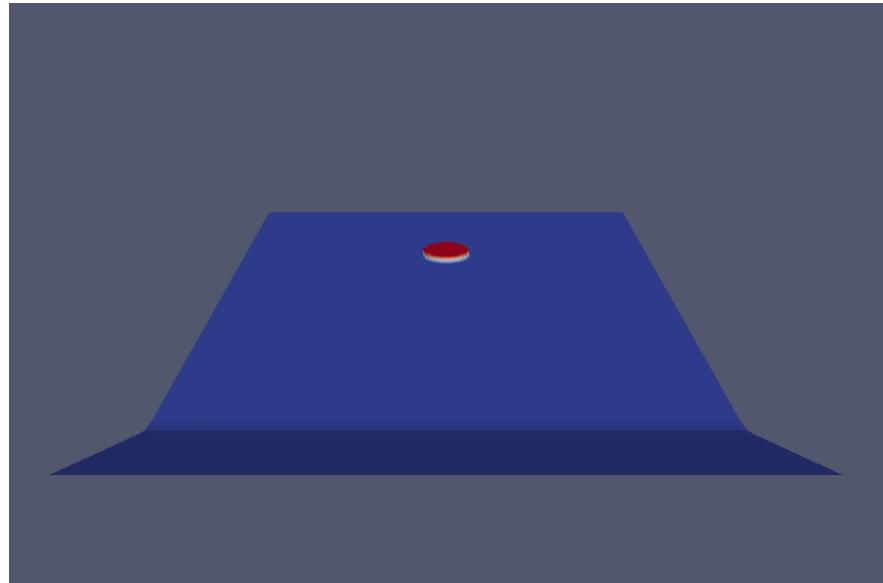


Figure 6: Initial configuration of the pile on the incline

4.1.1 Heuristic Method

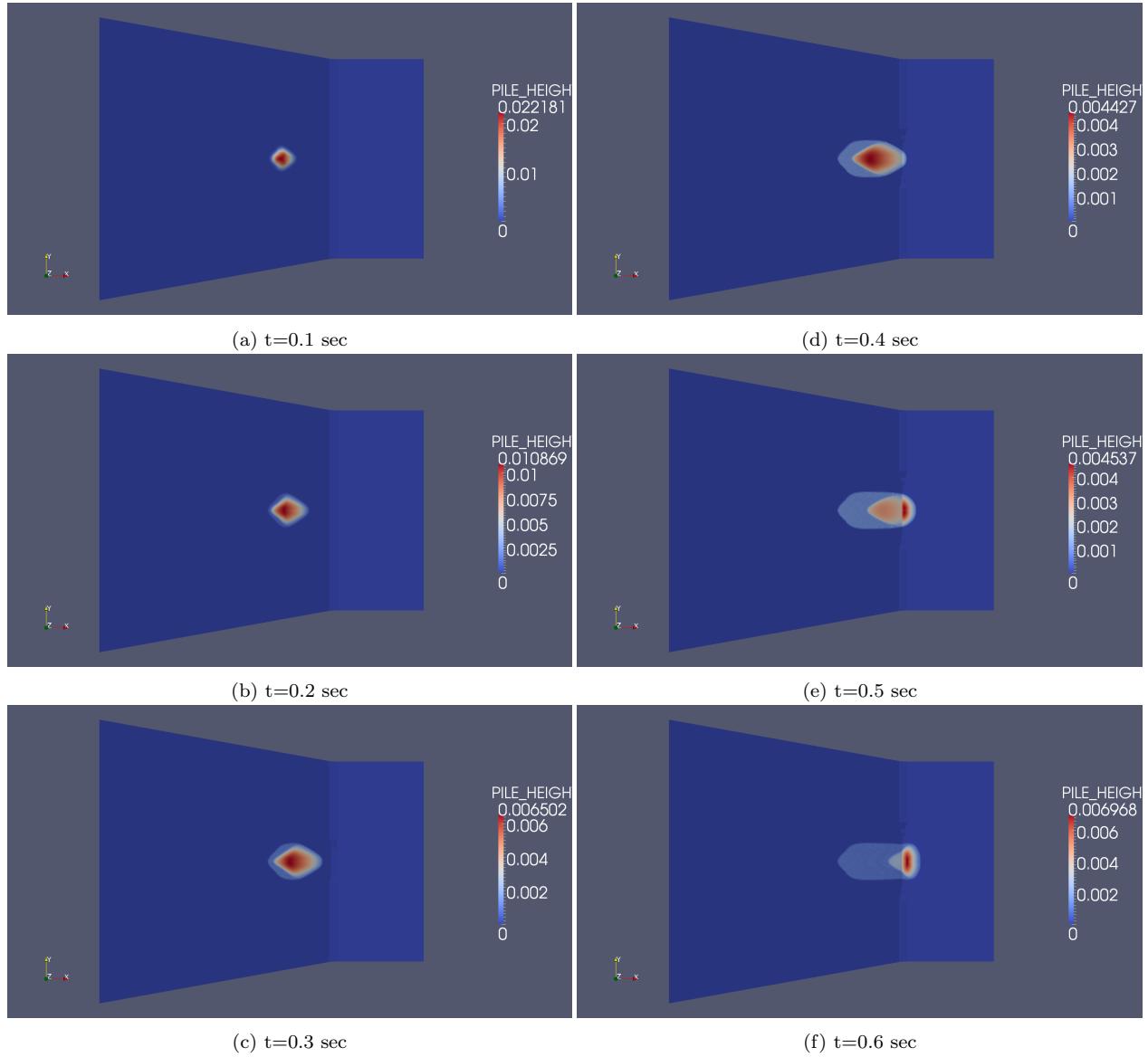


Figure 7: Pile height contour and interface location at different time steps

4.1.2 Phase field Method

In this section, the interface tracking results for a pile of volcano moving downward on an incline is displayed.

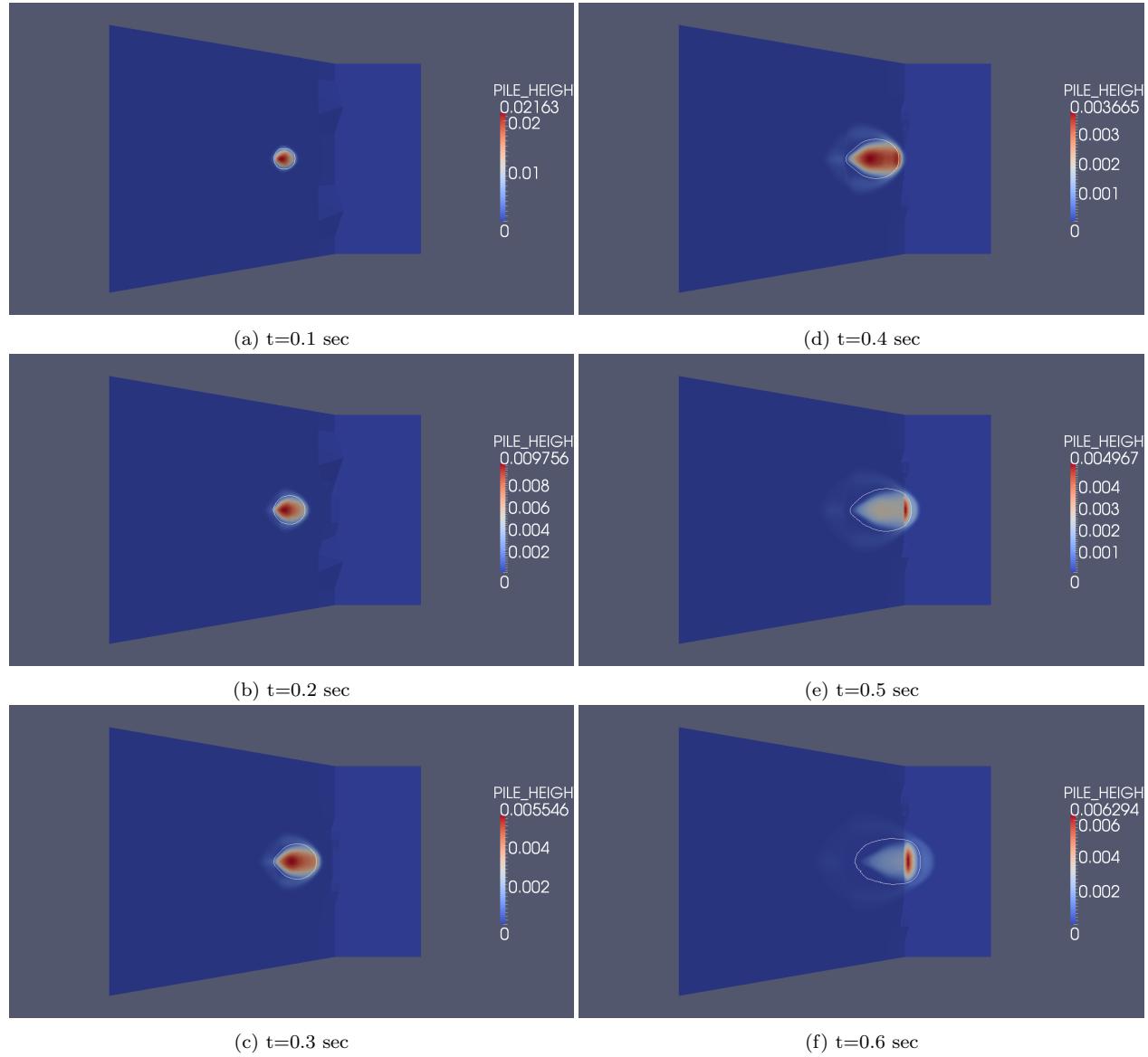


Figure 8: Pile height contour and interface location at diffeternt time steps

4.1.3 Level set Method

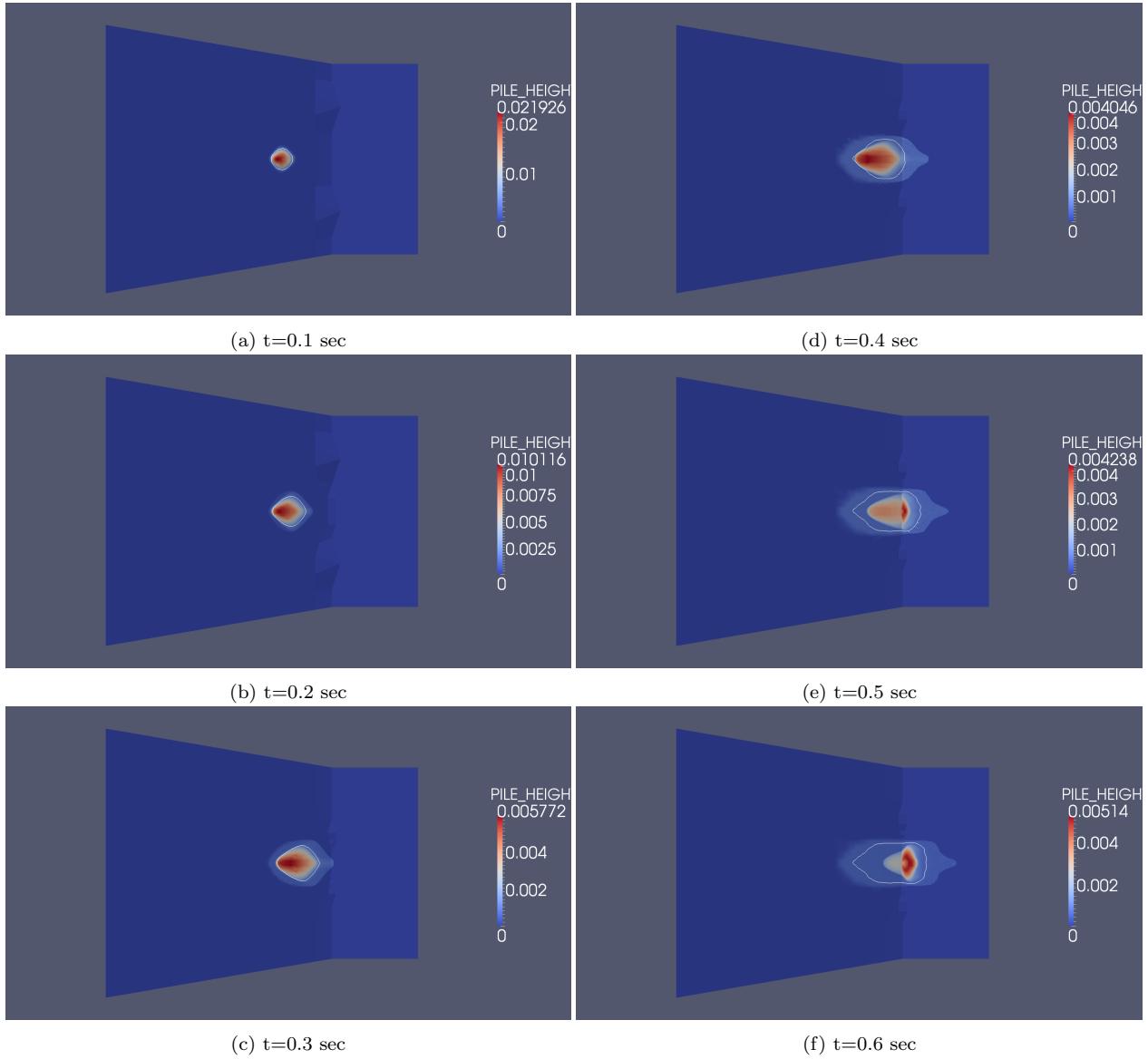


Figure 9: Pile height contour and interface location at diffeternt time steps

4.1.4 Comparison

For quantitative comparison of the results, we compared different schemes on three measurable quantities. The first measure is the extent of pile in X direction, the second one the extent of pile in Y direction, and the last one is the area of pile.

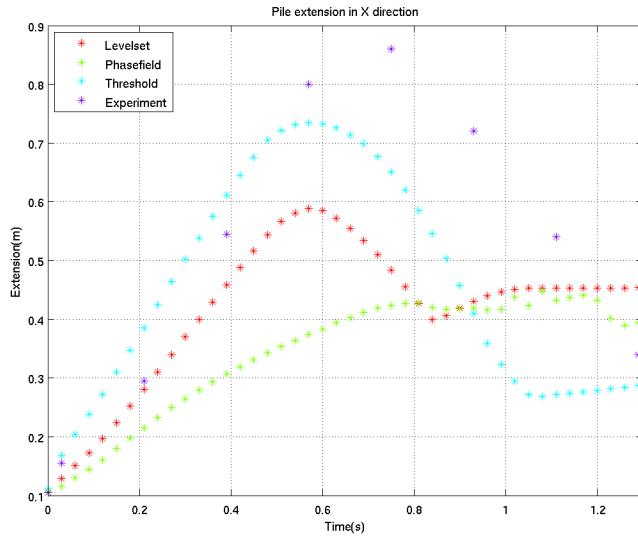


Figure 10: Extension of pile in X direction

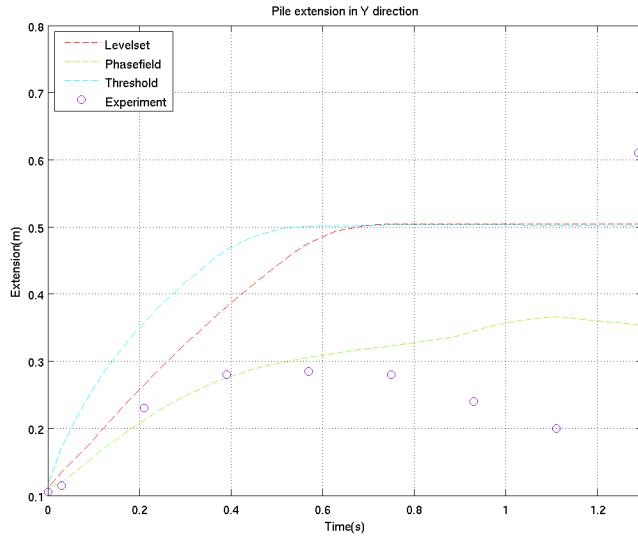


Figure 11: Extension of pile in Y direction

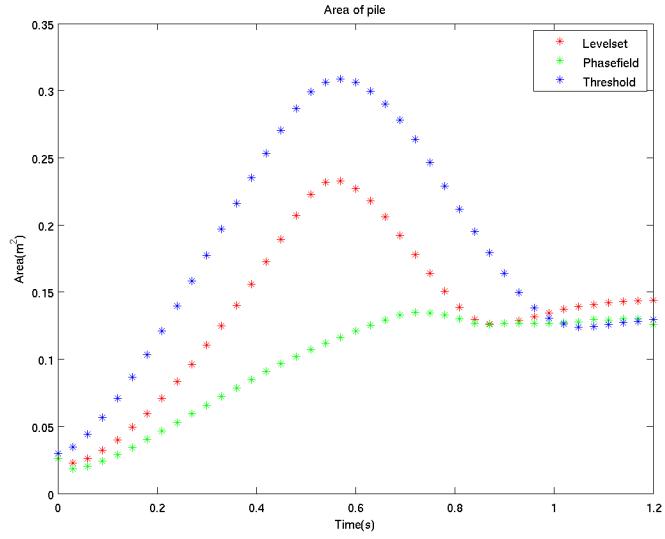


Figure 12: Area of pile

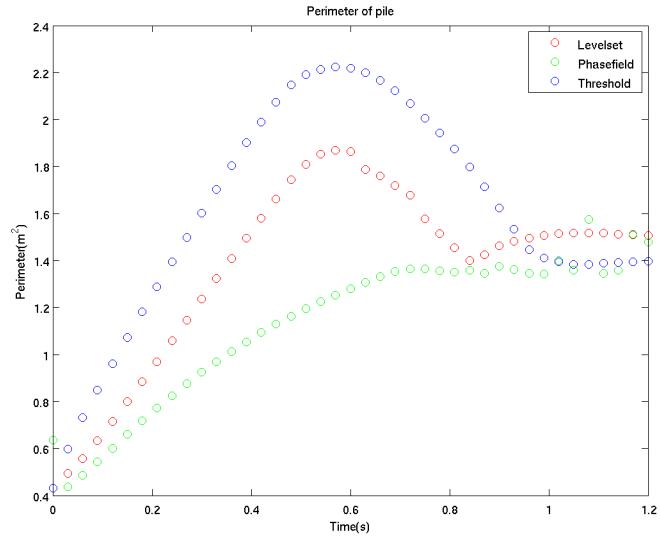


Figure 13: Perimeter of pile

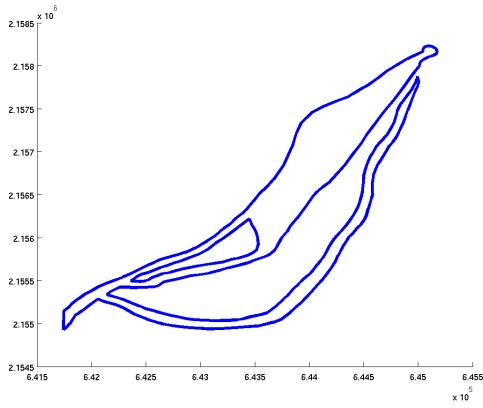
In these figures, the DG is the result of Dicountinouse Galerkin (DG) version of TITAN2D [1], and the experiment is the result of experiment that is reported in [2].

4.2 Colima Volcano

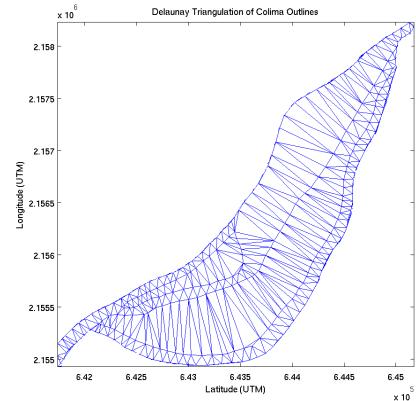
In this section, we verify our different interface capturing methods with a field data of an eruption of Colima volcano. Colima volcano in Mexico is one of the most active volcano in North America, and this eruption has happened in 16-17 April 1991 [1]. The topography of Colima volcano is such that small changes the initial location of the pile leads to a completely different path of flow, so a good performance of any of the interface capturing methods for this case promises a reliable method for other volcanoes. To be able to compare our results with the outline of deposit of flow, we record the history of all points with the finest resolution of the mesh during the simulation to check the points that are placed exactly on the boundary of the flow. As a definition boundary here means that the points that the flow passed from them, but has not passed one of their neighbors in different directions. For phase field and level set just with recording the maximum absolute value of phi we can find these points by plotting $\phi = 0$, this contour produces deposit line of flow during the time. For the heuristic method we recorded the minimum of the pile height during time and plotted the contour of $h = h_{scale} \times GEOFLOWTINY$, where $h_{scale} = Volume^{\frac{1}{3}}$.

The resolution of the digital elevation model (DEM) of Colima volcano that here we used is 5 meter which is the finest DEM that we have ever used for this volcano.

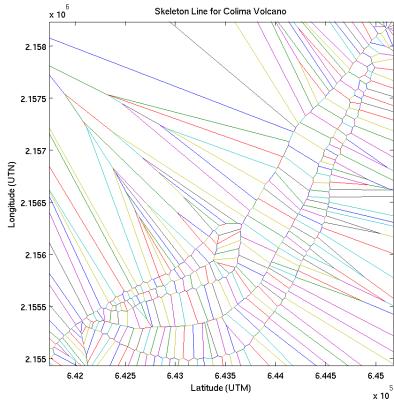
The outline of this eruption is available in [?], we used the method that described in [?] to extract the skeleton line of the flow to be able to compare the results quantitatively. The procedure of the extraction of the skeleton line is shown in the following pictures.



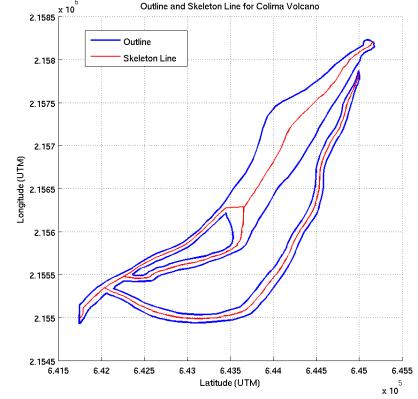
(a) Outline of flow from field data



(b) Delaunay triangulation based on the outline of the flow



(c) Finding the middle of the edges of the triangles that connects the outline of the flow



(d) Connecting the middle verticals that are inside of the outline of the flow, in this picture the outline of the flow (blue line) and the skeleton line of the flow (red line) can be seen

Figure 14: In this set of pictures the steps that have to be taken for finding the outline of the flow is displayed respectively

After finding the skeleton line, we mapped the outline of the flow and the skeleton line on the Colima volcano using KML language and Google earth application 15. In pictures ??- the result of each methods and their comparison can be seen.

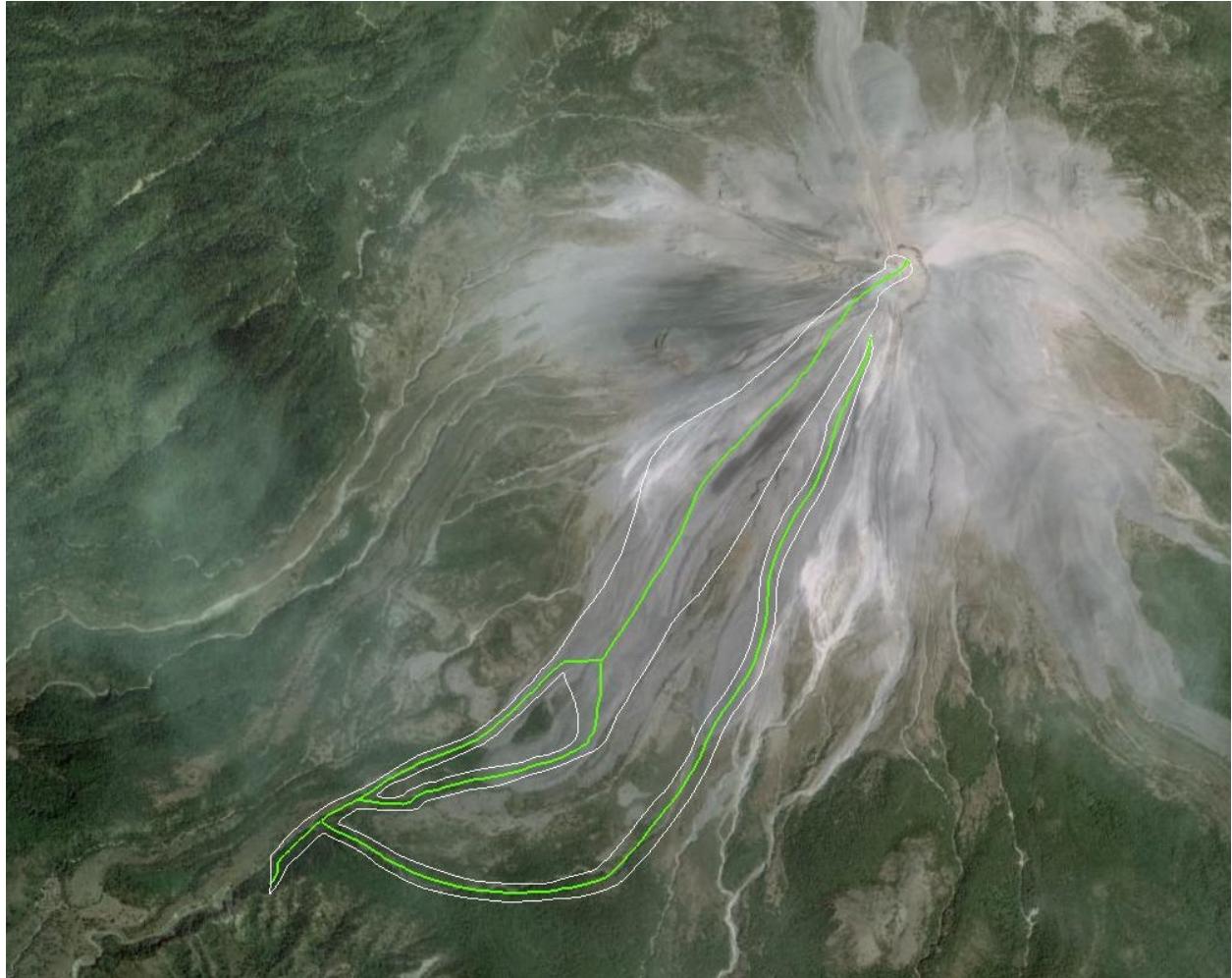


Figure 15: Skeleton line (white line) and outline of the deposit of eruption 1991 (yellow line)

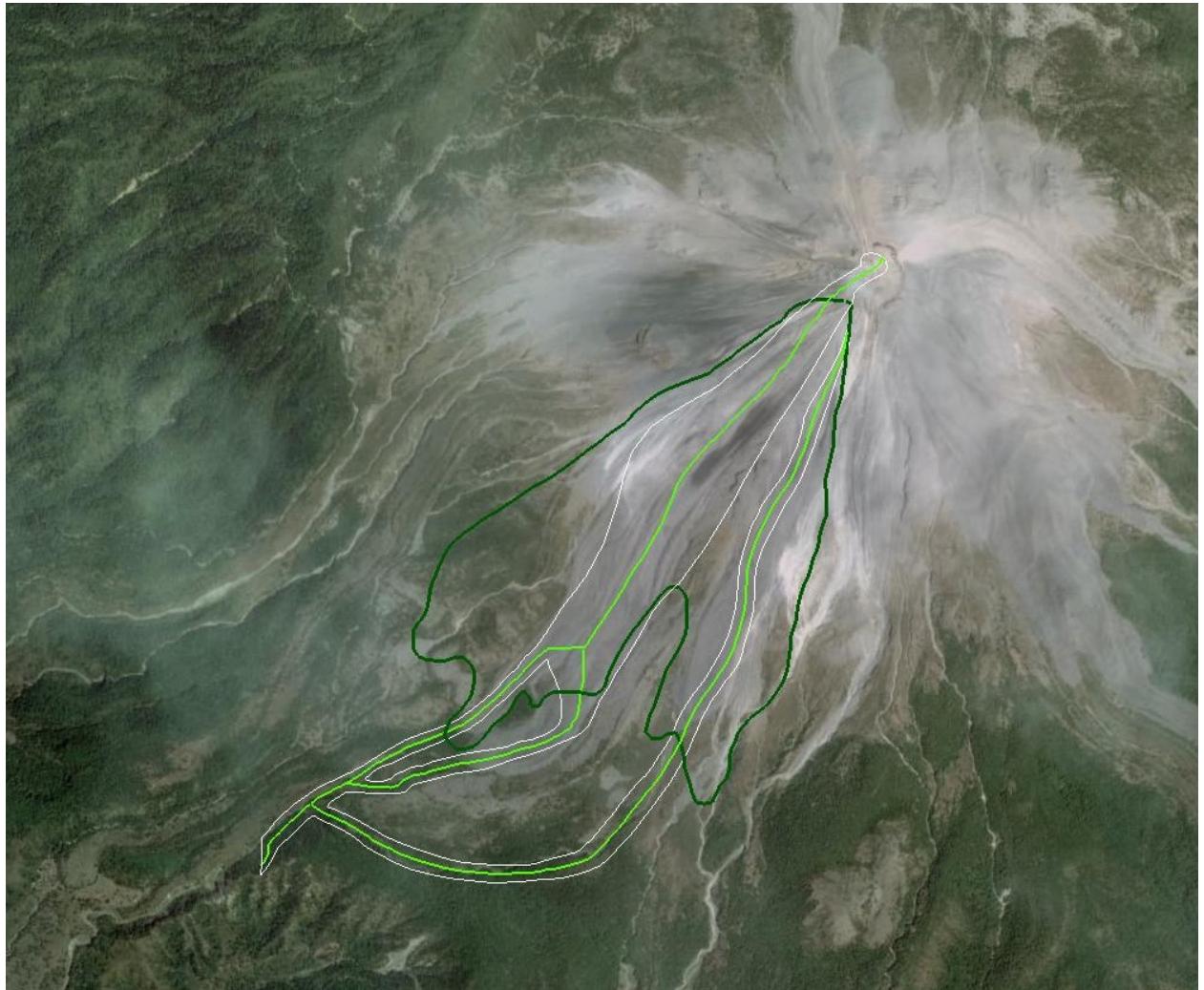


Figure 16

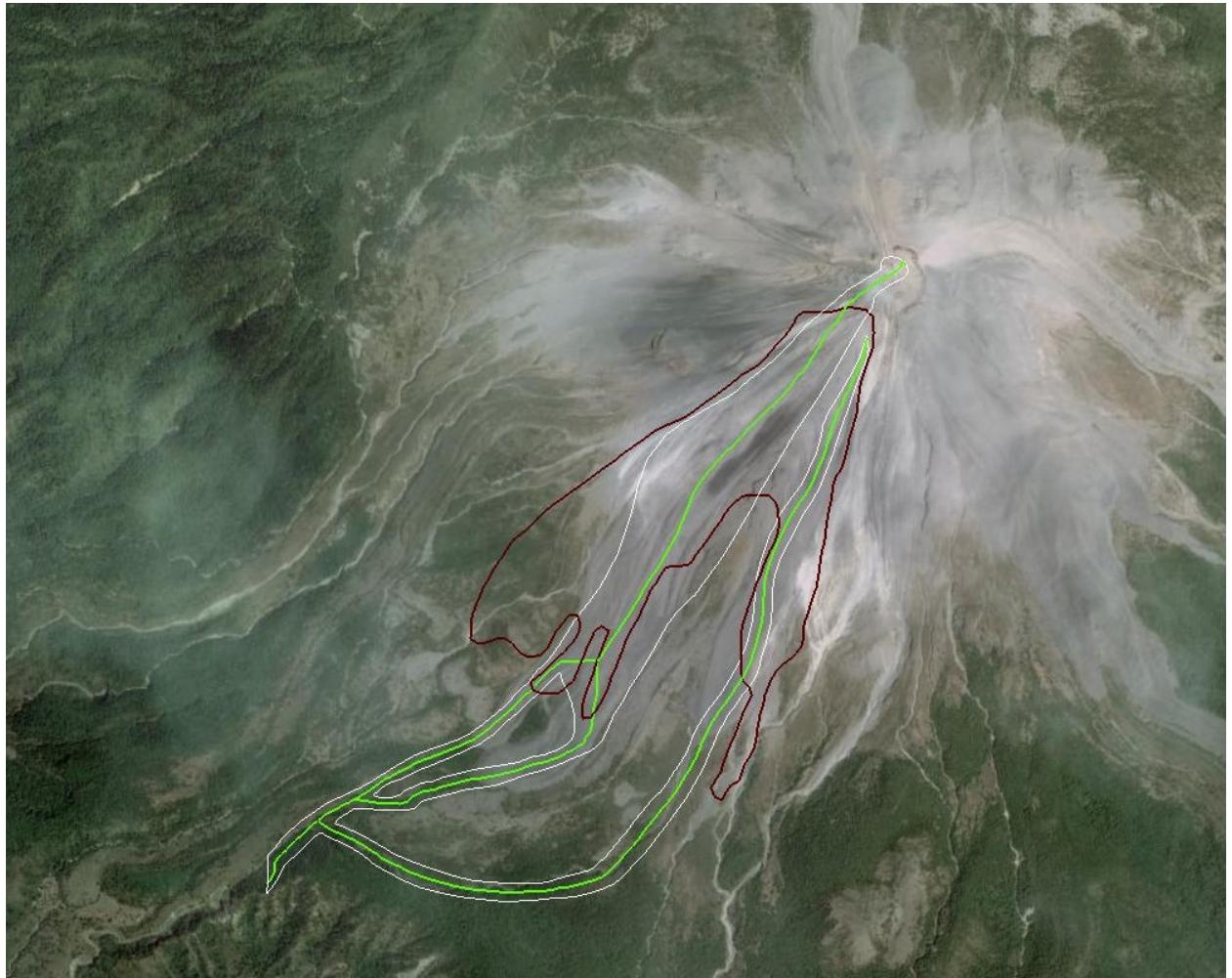


Figure 17: Skeleton line (white line) and outline of the deposit of eruption 1991 (yellow line)

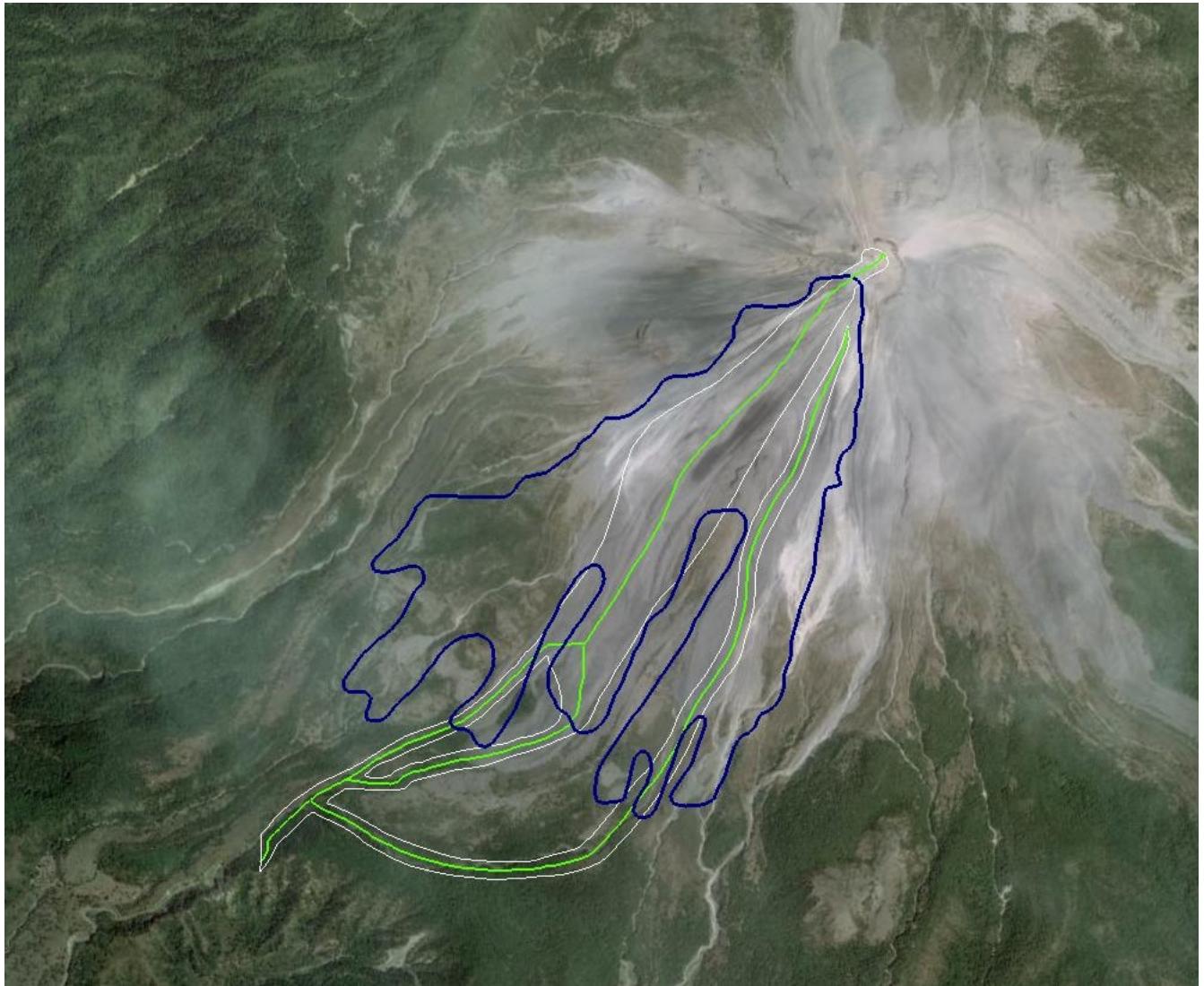


Figure 18: Skeleton line (white line) and outline of the deposit of erution 1991 (yellow line)

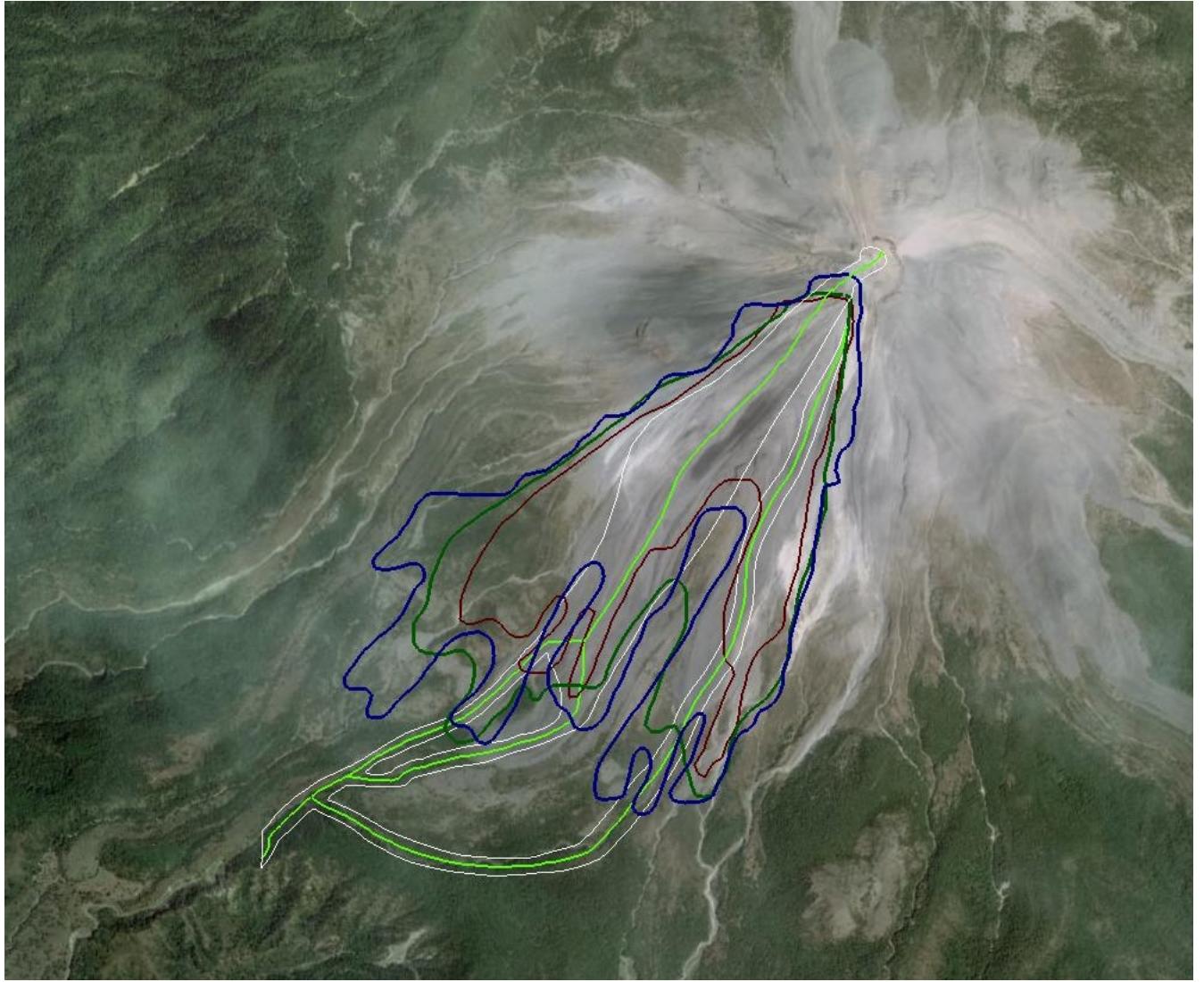


Figure 19: Skeleton line (white line) and outline of the deposit of eruption 1991 (yellow line)

5 Conclusions

The numerical solution of the Savage-Hutter (and similar “shallow-water”) equations have historically been plagued by several interrelated numerical difficulties which are collectively characterized by a non-physically thin-layer extending large distances from the realistic main body of the flow. In the best case, this “thin-layer problem” means a “no flow” boundary line must be arbitrarily drawn at some given depth contour. In the worst case, it can cause severe numerical instability that prevents any simulation of a particular event. In this paper, we have described some features of the thin-layer problem, some underlying causes that are

common to virtually all numerical solution methodologies. Moreover, we have presented a heuristic method and compared two interface capturing approaches that mitigate this problem by addressing its root causes. We implemented these thin-layer control strategies in Titan2D, our high performance finite volume solver of the depth-averaged granular flow equations. Numerical simulations were performed for geophysical mass flows at two separate locations.

The numerical experiment were conducted for an inclined plate that finally continues horizontally. This case was tested with all of the approaches, which not only prevented the loss of numerical stability but also demonstrated behavior that is, at least qualitatively, consistent with expectations.

The second location, Colima volcano, which is an active volcano in Mexico, was selected on the basis that the DEM had provoked thin-layer numerical difficulties from an earlier version of Titan2D, and that a campanologist familiar with the location had selectively tuned Titan2D's few input parameters and also computational and post-processing thresholds to produce results that closely matched reality for that location. The new version of Titan2D, which implemented our thin-layer control strategy, automatically (i.e. without tuning) reproduced a flow outline that had even greater agreement with the historical data.

Then the results of these approaches were compared. The result of this comparison showed a very good consistency between these approaches.

On the basis of these very positive results, we concluded that our thin-layer control strategy, and interface capturing approach provides sufficient benefit. While all of these approaches to thin-layer mitigation was developed in the context of Titan2D's capabilities, much of it should be appropriate for use in depth-averaged flow solvers with different numerical implementations.

References