

# 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 (SW) equations (or similar granular flow's equations), and has been the focus of much research effort. In this paper, to the best knowledge of the authors, for the first time we used level set and phase field methods to solve this issue and other related problems. We also proposed a new heuristic method to address this problem. We implemented all of these methods in TITAN2D which is a parallel adaptive mesh refinement toolkit designed for numerical simulation of granular flows.

The results of the methods were presented for simple inclined plane and Colima volcano. For the inclined plane we verified the results with experimental data and for Colima volcano they have been compared with field data. We successfully captured the interface of the flow and solved the accuracy and stability problems related to thin layer problem in SW numerical solution. The comparison of results shows that although all of the methods can be used to address this problem, and each of them has its own advantage/disadvantages that has to be chosen carefully depend to the criteria of the interested case.

**Keywords:** Shallow water flow, Thin layer, Wetting/Drying, Phase field, Level set

## 1 Introduction

Shallow water (SW) flows include a wide range of fluid flows. In SW flows the fluid depth is much smaller than the characteristic length of the fluid body. In deriving the corresponding governing equations, the shallowness of flow allows us to neglect the variation of state variables in perpendicular direction to the basal surface, and the effect of this replaced with an integrated average in the equations [43]. This mathematical simplification allows us to simplify three dimension flow in two dimension. The aforementioned condition holds for many circumstances in geophysical flow and the same conservation equations with some minor modifications could be used to study this type of flow.

To this aim, Eglit and Sveshnikova [21] modified the depth-averaged Saint Venant equations to simulate granular snow avalanches, and almost a decade later Savage and Hutter [43] adapted the depth averaged Saint Venant equations to accommodate geophysical mass flow. Since this type of flow has free moving boundaries, one of the basic difficulties of the numerical solution, is identifying the location of flow interface. More clearly, the governing equations are just valid in wet areas, so we need a strategy to discriminate between wet and dry areas in the numerical simulation. This problem is usually called the **wetting and drying (WD)** or **Thin-Layer** problem in SW context. In our previous work [4], we showed how the speed of wave has to be modified for flow near the vacuum region based on [47] to mitigate the stability problem, but lack of a solution for spreading of the flow leads to the formation of a non-physical thin 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 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, what is the best numerical solution for the location of the front of the flow? Does the flow reach to a specific location? What is the distance of high risk locations from the infrastructures? The answer of all the above questions is not possible without good information about the interface of the

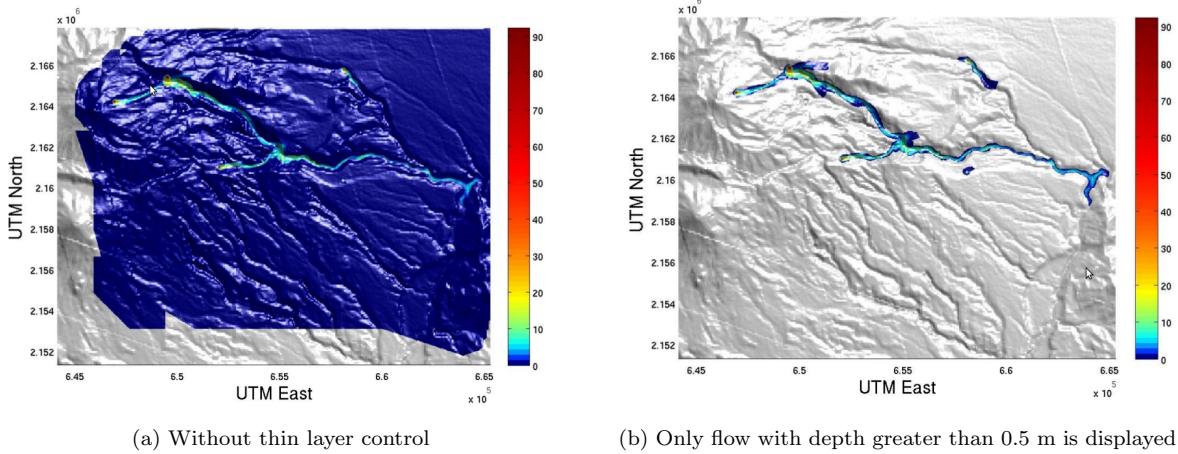


Figure 1: Thin layer problem in maximum over time flow depth simulation of the 1955 debris flow at Atenquique, Mexico [19].

flow during its traveling path.

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

1. Non-physical fast numerical flow speed.
2. Non-physical thin layer (orders of magnitude thinner than a grain of sand). This results from the combination of conservation of mass and the numerical wicking away of material.
3. Non-physically high flow speeds. Since the solution of SW equations are momentum,  $\{hV_x, hV_y\}$ , to find velocities,  $\{V_x, V_y\}$  used in solution process, the momentum must be divided by 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.

Figure 1 is an example that demonstrate this problem for numerical simulation of granular flow in Atenquique which is a village near Colima volcano in Mexico. The left figure shows the numerical solution of flow height without using any control for thin layer problem, and the right figure shows the same result with using a naive control with plotting the regions with flow height grater than 0.5 meter. As discussed above, if we do not use any control on the numerical solution, the spread thin layer causes instability and inaccuracy in the obtained results.

## 1.1 Review

A review of the literature shows numerous attempts were made to overcome this problem over time especially in the recent years [35, 8, 6, 11, 15, 31, 20, 14]. Here we will briefly review the topic and refer the interested reader to available references 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. In this

approach, the reconstruction could be also with taking to account 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 both tracking and capturing is that they are usually used for Lagrangian and Eulerian methods respectively. For more information about this group of methods please see [24, 49, 37, 5, 28].

In Heuristic methods, the idea is to mitigate thin layer problem with some heuristic strategy. This strategy could be using a threshold to control the thin layer, filling the entire domain with a shallow level of fluid, reconstructing the flow, adjusting the flux, or any combination of them. Most of the attempts that are made to address the thin layer problem are categorized under this sort of method [6, 11, 14, 31].

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 view, 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 represents 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 (MAC) [27], Simplified Mark And Cell (SMAC) [18], and Surface Marker [50] methods are some examples of Lagrangian interface tracking techniques. 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. Tai et al. [46] 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.

In the current paper we focus more on interface capturing or Eulerian methods. Level set, phase field and Volume of Fluid (VOF) are very 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 [28] 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 [52] 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 [23] and Gopala and van Wachem[25].

Phase field is an Eulerian interface capturing scheme that the interface is implicitly related to the order parameter which shows the phase variation on the domain [5]. In this method the interface is a diffusive region between the phases in contrast to level set which yields to a sharp interface between the phases. Based on our best knowledge no published work used earlier this method for capturing the interface in SW. Cahn-Hilliard and Allen-Cahn formulations are two principal formulation for this method [13, 12, 51]. 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 later form much easier to implement.

The Level set method is another Eulerian interface capturing technique. It was introduced by Osher and Sethian in 1988 [37]. In this method, the scalar variable is a signed distance function which shows the distance of each point to the interface. The only work that used level set method for WD problem is Quecedo and Pastor [42]. They developed Taylor-Galerkin approach to solve the SW equations. 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.

## 1.2 Our Contribution

In this paper, our approach to address the resulted problems from the thin layer problem (problem 1-4 discussed in introduction section), consists of two parts. The first part is to find the interface of the flow, and the next part is to do some steps based on the result of the first part to resolve the aforementioned issues. To capture the interface we study three different methods. The first one is to use a very small dimensionless threshold, which we call it GEOFLOW\_TINY, to distinguish between wet and dry areas. The other method that we implement to capture the interface is phase field.

## I have to rewrite this part

that is resulted from the thin layer problem and discussed in two do some we proposed a new multifaceted Heuristic approach and also tested the effectiveness of Level set and Phase field methods to mitigate the thin layer problem difficulties. Our approach in this work consist of two parts. for numerical solution of SW equations. To the best knowledge of authors, the two later methods are for the first time applied to geophysical flow. Although these methods have been used for granular type SW flows, most of the conditions holds and the results are applicable for other type of SW flows. We verified the numerical results with experimental result of a granular SW flow over an inclined plane ends to a horizontal surface , as well as the field data for Colima volcano. In Eulerian interface capturing methods we tried to stick to the basic methods, and decrease the computational cost ,when possible, with using some techniques.

In the heuristic approach we 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 level set part, we followed the instructions in paper[45] with the re-initialization scheme introduced in Adalsteinsson’s work [3]. To solve both of these hyperbolic equations we used a first order accurate solver.

For phase field method, we chose Allen-Cahn formulation. This formulation is not mass conservative. To satisfy mass conservation constrain in this formulation, a Lagrange multiplier is added to RHS of the equation. In this work, we derived a Lagrange multiplier for SW equation. To decrease the computational and memory cost, we used operator splitting and matrix free techniques (detail in section 3.2).

As a general comparison between the methods it could be mentioned that the first approach is less expensive, but the obtained interface moves a little ahead of the interface found in experimental and field data is. On other hand, the interface capturing methods have more strong mathematical and physical structure, and can be coupled with the conversations equations, but are computationally more expensive, and harder to implement.

The structure of the rest of this paper is as follows: in next part SW equations for the geophysical flows and the common features of the solver that we used for all of the three methods are introduced. In section 3, different methods that were employed to mitigate WD problem 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 [32, 29, 26, 30, 41, 44]. In our earlier work [40, 38, 39], we developed the Titan2D depth-averaged geophysical flow simulator. Titan2D is

parallel with dynamic re-partitioning, 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:

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

In these equations:

- The coordinate system (see Figure 2) is aligned such that  $x$  and  $y$  are directions tangential 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  $k_{ap}g_z \frac{h}{2}$  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 friction that resists the internal shearing motion of the material see eq. (3).

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

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 Harten-Lax-Van Leer (HLL) [48] Riemann solver that is explained in the next section.

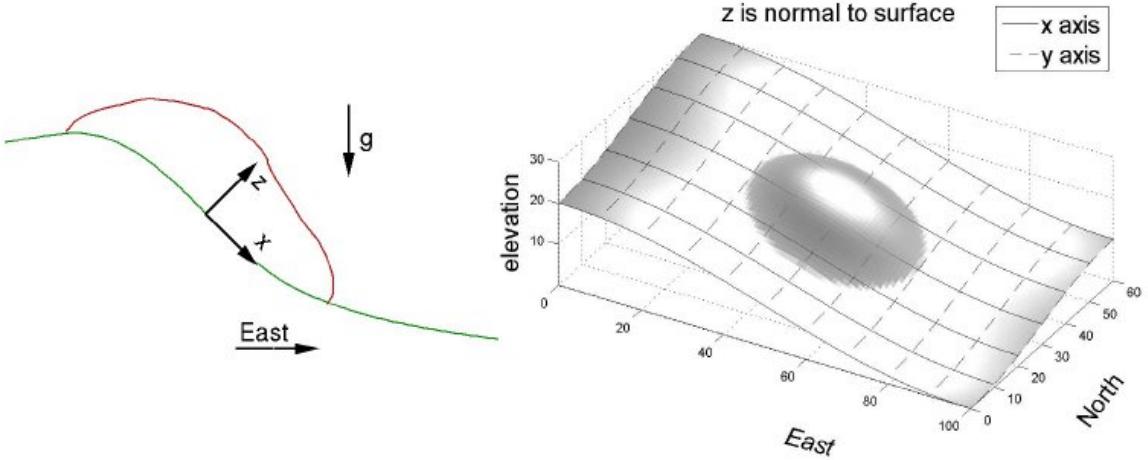


Figure 2: In the local coordinate system the  $z$  direction is normal to the surface; the  $x$  and  $y$  directions are tangential to the surface.

## 2.1 Using Appropriate Riemann Fluxes

For flux computation TITAN2D uses a 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 [47], 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 [22] 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 3.

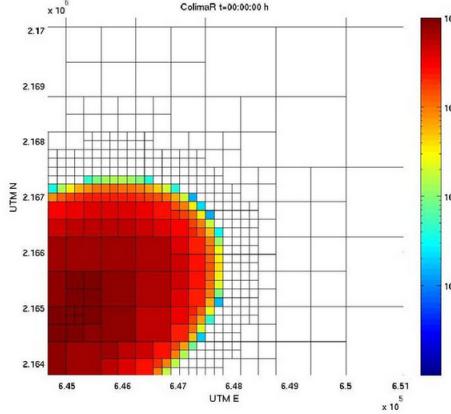


Figure 3: Buffer cells

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 near the boundary.

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. In solution section, we explain how we select the elements near the boundary for refinement for each method of interface capturing that we use. The rings are a number of cells wide equal to or greater than the number of iterations between mesh adaptations (we are not doing mesh adoption at each time step). This guarantees that material will only flow into maximally refined dry cells at the wet-dry front. here I have to rewrite, I have to say we use GEOFLOW\_TINY in all of the methods, I have to also add an adative section to each part of solution 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 1. 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.

### 3.1 First: Heuristic methods

Usually, heuristic methods use one or combination of following four strategies [35]:

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 3.1 compares them very briefly.

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

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[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. [46] 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.

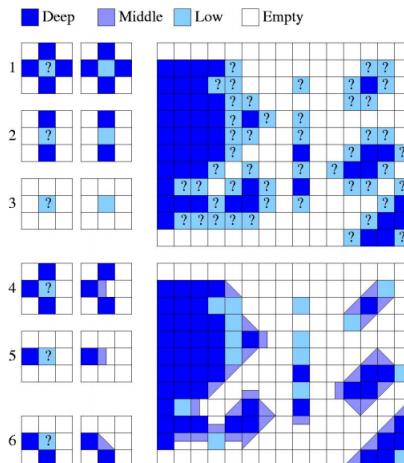


Figure 4: Interface Reconstruction in TITAN2D

Specifically, each partially wet cell is assumed to be split by a straight line into a completely dry and a 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` threshold<sup>1</sup>. 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 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.

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.  $a = \sqrt{k_{apgh} \frac{A}{A_{wet}}}$ . 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 with, 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,  $\Delta t_{dry}$  is the portion of the time-step for which the edge is completely dry,  $\Delta t_{part}$  is the portion of the time-step for which the edge is partially wet and partially dry,  $\Delta t_{wet}$  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).

---

<sup>1</sup>As 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.

### 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  $\varphi$ , 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 [17, 5, 9, 33] are good references about the history and evolution of the method.

Phase diffusion methods and particularly phase field method is based upon the notion that the interface between phases is a diffusive region rather than a sharp interface with a finite width. Value of  $\phi$  is constant inside the bulk phases but changes smoothly between the phases. In this work  $\phi$  is 1 for the fluid phase  $\{\mathbf{x}, \phi(\mathbf{x}, t) = 1\}$ , and it is -1 for void regions  $\{\mathbf{x}, \phi(\mathbf{x}, t) = -1\}$ , and is between -1 to 1 on the diffusion region  $\{\mathbf{x}, -1 < \phi(\mathbf{x}, t) < 1\}$ . Then we can implicitly assume the interface of flow is where  $\{\mathbf{x}, \phi(\mathbf{x}, t) = 0\}$ .

Cahn-Hilliard and Allen-Cahn are the two principal phase field formulations. Basically, Cahn-Hilliard and Allen-Cahn are two possible gradient flows that can describe the phase diffusion. Both of the formulation can be written in the form of equation (8). As could be seen, the time material derivative of  $\varphi$  is related to the gradient of a energy functional. For Allen-Cahn formulation, this energy functional is Ginzburg-Landau (9). The basic difference between this formulations is that Cahn-Hilliard is mass conservative but Allen-Cahn is not. From computational point of view, there is a bi-harmonic term in the right hand side of Cahn-Hilliard equation which is forth order derivative and makes computationally expensive to implement. Usually, people try to use Allen-Cahn first and then if the result was not satisfactory they switch to Cahn-Hilliard phase field. Since in our work we intend to find the interface with the least computational cost Allen-Cahn form was chosen.

$$\frac{D\varphi}{Dt} = -\frac{\gamma}{\eta} \frac{\delta W}{\delta \varphi} \quad (8)$$

$$W(\varphi, \nabla \varphi) = \int_{\Omega} \left\{ \frac{\eta}{2} |\nabla \varphi|^2 + F(\varphi) \right\} dx \quad (9)$$

In equation (8),  $\eta$  is a constant that regulates the capillary width or diffusion width, and  $\gamma$  denotes elastic relaxation constant. In our case, we select:

$$F(\varphi) = \frac{1}{4\eta} (\varphi^2 - 1)^2, \quad (10)$$

which is the well-known double-well potential function, and represents the interactions of different volume fractions of individual species. The gradient part  $\frac{\eta}{2} |\nabla \varphi|^2$ , is the relaxtion part that relates the mass average of the energy, with the volume average of the elastic energy [10, 34]. Putting (10) in to equation (8) leads to:

$$\frac{\partial \varphi}{\partial t} + \vec{V} \cdot \nabla \varphi = \gamma(\Delta \varphi - F'(\varphi) + \xi(t)), \quad (11)$$

where

$$F'(\varphi) = \frac{\delta F}{\delta \varphi} = \frac{1}{\eta} \varphi (\varphi^2 - 1). \quad (12)$$

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 equation (11) by  $\xi(t)$ . In our case, we have an initialie pile of volcanicc flow that has be constant during the entire simulation, so the constraint that we want to be satisfied is:

$$\frac{D}{Dt} \int_{\Omega} \varphi h \, dx = 0. \quad (13)$$

The reason that equation (13) holds is that we selected  $\{\mathbf{x}, \phi(\mathbf{x}, t) = 1\}$  for the regions that granular material exist. Thus, the result of integral is nothing else than the initial volume of pile that has to be fixed for all

simulation times. This constraint is different than the constraint that usually is used to conserve the mass in Allen-Cahn formulation, and we have to find a new  $\xi(t)$ . The details of the derivation is in Appendix A, and the result of that is shown in (14).

$$\xi(t) = \frac{1}{V_0} \int_{\Omega} hF'(\varphi) dx \quad (14)$$

where  $V_0$  is the initial volume of the pile. In all of our simulation we used  $\eta = 4\delta x$ , where  $\delta x$  is equal to the smallest cell length. Since we use the result of phase field for mesh refinement, and we always have the smallest cell size near the interface, using this  $\eta$  means that we select the length of capillary width to be equal to length of four cells.

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. Since phase field method has a second order derivative, to satisfy CFL condition which in this case is related to the inverse square of the spatial increment,  $CFL \propto \Delta x^{-2}$ , thus time increment must be proportional to the square of the spatial increment,  $\Delta t \propto (\Delta x)^2$ , makes the time step too small and causes unaffordable 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 both make the time scheme stable for larger time steps, and keep the computational cost low. By the operator splitting scheme that we used here, the time integration of laplacian term is performed with implicit Eulerian scheme and the rest of terms are updated in the explicit time integrator. This solution allows us to select larger time steps for the 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. Equations (16), and (15) are respectively implicit and explicit time integrators.

$$\begin{aligned} \frac{\partial \varphi_1}{\partial t} + \vec{V} \cdot \nabla \varphi_1 &= f(\varphi_1) & \frac{\partial \varphi_2}{\partial t} &= \alpha \nabla^2 \varphi_2 \\ L_1 = V_x \frac{\partial}{\partial x} + V_y \frac{\partial}{\partial y} & & L_2 = \alpha \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) & \\ \frac{\varphi_1^{n+1} - \varphi_1^n}{\nabla t} &= f(\varphi_1^n) - L_1 \varphi_1^n & \frac{\varphi_1^{n+1} - \varphi_1^n}{\nabla t} &= \alpha L_2 \varphi_2^{n+1} \\ \varphi_1^{n+1} &= \nabla t (f(\varphi_1^n) - L_1 \varphi_1^n) & (15) & \\ (1 - \nabla t \alpha L_2) \varphi_1^{n+1} &= \varphi_2^n & (16) & \end{aligned}$$

To preserve the scalability of TITAN2D, for solving the equations we used PETSc [7] library. GMRES solver, which is a iterative Krylov subspace solver was exploited to solve the system of equations. To decrease the memory cost, we used a 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 methods 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 saving in required memory cost.

### 3.3 Third: Level Set Method

The last method that we use to capture the interface for a SW flow is the Level set method. The Level set method is another Eulerian interface capturing method; it was introduced by Osher and Sethian in 1988 [37]. The basis of the method is to capture the interface by means of solving a hyperbolic Hamilton-Jacobi PDE on the computational domain which follows 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 can be achieved. To derive the level set equation, given the initial location of the boundary an initial signed distance function on the domain which can 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  $\Psi(X, t)$  :

$$\frac{\partial \Psi}{\partial t} + F |\nabla \Psi| = 0 \quad (17)$$

Substituting  $F = \vec{V} \cdot \vec{n}$ , and  $\vec{n} = \frac{\nabla \Psi}{|\nabla \Psi|}$  relations into eq. (17) leads to (18)

$$\frac{\partial \Psi}{\partial t} + \vec{V} \cdot \nabla \Psi = 0 \quad (18)$$

Equation (18) is a non-conservative hyperbolic equation. We used a first order upwind scheme that is explained in detail in [] to solve this equation.

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \{F_{i+\frac{1}{2}}^- + F_{i-\frac{1}{2}}^+\} - \frac{\Delta t}{\Delta y} \{G_{i+\frac{1}{2}}^- + G_{i-\frac{1}{2}}^+\} \quad (19)$$

Where  $F$  and  $G$  are the flux terms and are coming from the flux term for in upwind scheme: I have to correct the following formula and add reference above

$$F_{i+\frac{1}{2}}^- = \begin{cases} \frac{v_r + v_l}{2} (v_r - v_l) & v_l < 0, \\ 0 & . \end{cases}$$

and

$$F_{i+\frac{1}{2}}^+ = \begin{cases} 0 & v_l > 0, \\ \frac{v_r + v_l}{2} (v_r - v_l) & . \end{cases}$$

and the same formulation can be used to find G fluxes.

#### 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 [37], but what we implemented here is the method presented by Sussman et al [45]. In this reinitialization technique another hyperbolic equation is solved to adjust the  $\Psi$  value in the domain:

$$\frac{\partial \Psi}{\partial \tau} + \text{sgn}(\Psi)(|\nabla \Psi| - 1) = 0 \quad (20)$$

In eq. (20)  $\tau$  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 [2] was used to obtain the solution of re-initialization equation. Since the re-initialization is not required for every time step, the re-initialization was performed after every five time steps.

$$\Psi_{ijk}^{n+1} = \Psi_{ijk}^n - \Delta t \left( \max(F, 0) \nabla_{ijk}^+ + \min(F, 0) \nabla_{ijk}^- \right), \quad (21)$$

where

$$\begin{aligned} \nabla_{ijk}^+ &= [\max(D^{-x}\Psi_{ijk}^n)^2 + \min(D^{+x}\Psi_{ijk}^n)^2 \\ &\quad \max(D^{-y}\Psi_{ijk}^n)^2 + \min(D^{+y}\Psi_{ijk}^n)^2]^{1/2} \end{aligned} \quad (22)$$

$$\begin{aligned} \nabla_{ijk}^- &= [\min(D^{-x}\Psi_{ijk}^n)^2 + \max(D^{+x}\Psi_{ijk}^n)^2 \\ &\quad \min(D^{-y}\Psi_{ijk}^n)^2 + \max(D^{+y}\Psi_{ijk}^n)^2]^{1/2} \end{aligned} \quad (23)$$

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

## 4 Results

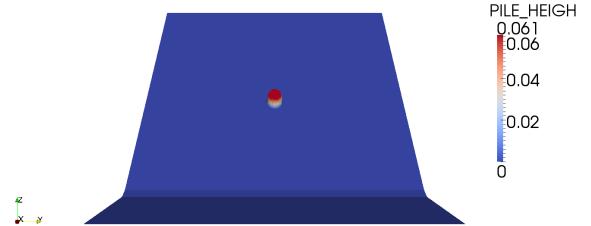
### 4.1 Inclined plane

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°



(a) Experimental setup



(b) Numerical simulation

Figure 5: Initial configuration of the pile on the incline

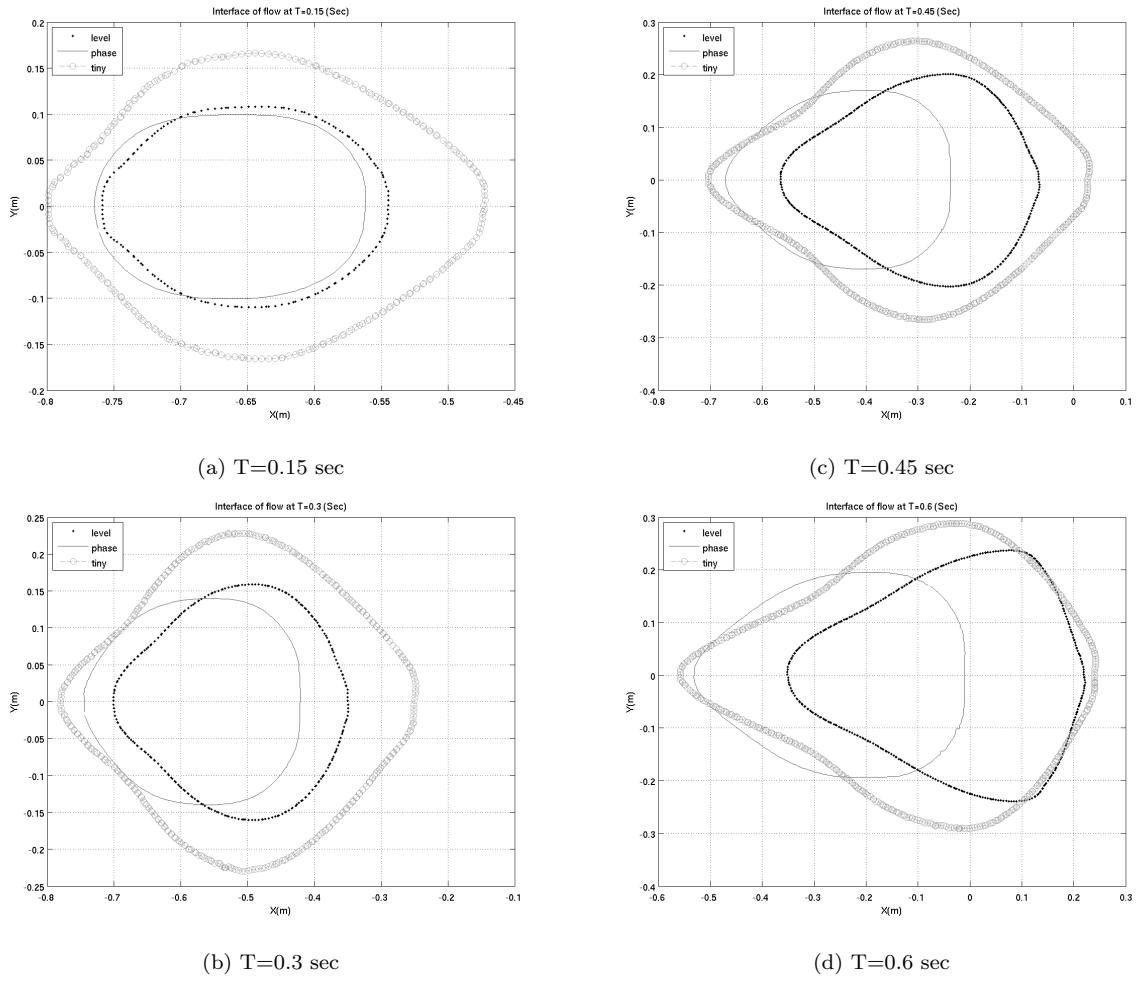


Figure 6: Pile height contour and interface location at different time steps

it would be very good if I can plot data of experiment for area and perimeter

#### 4.1.1 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.

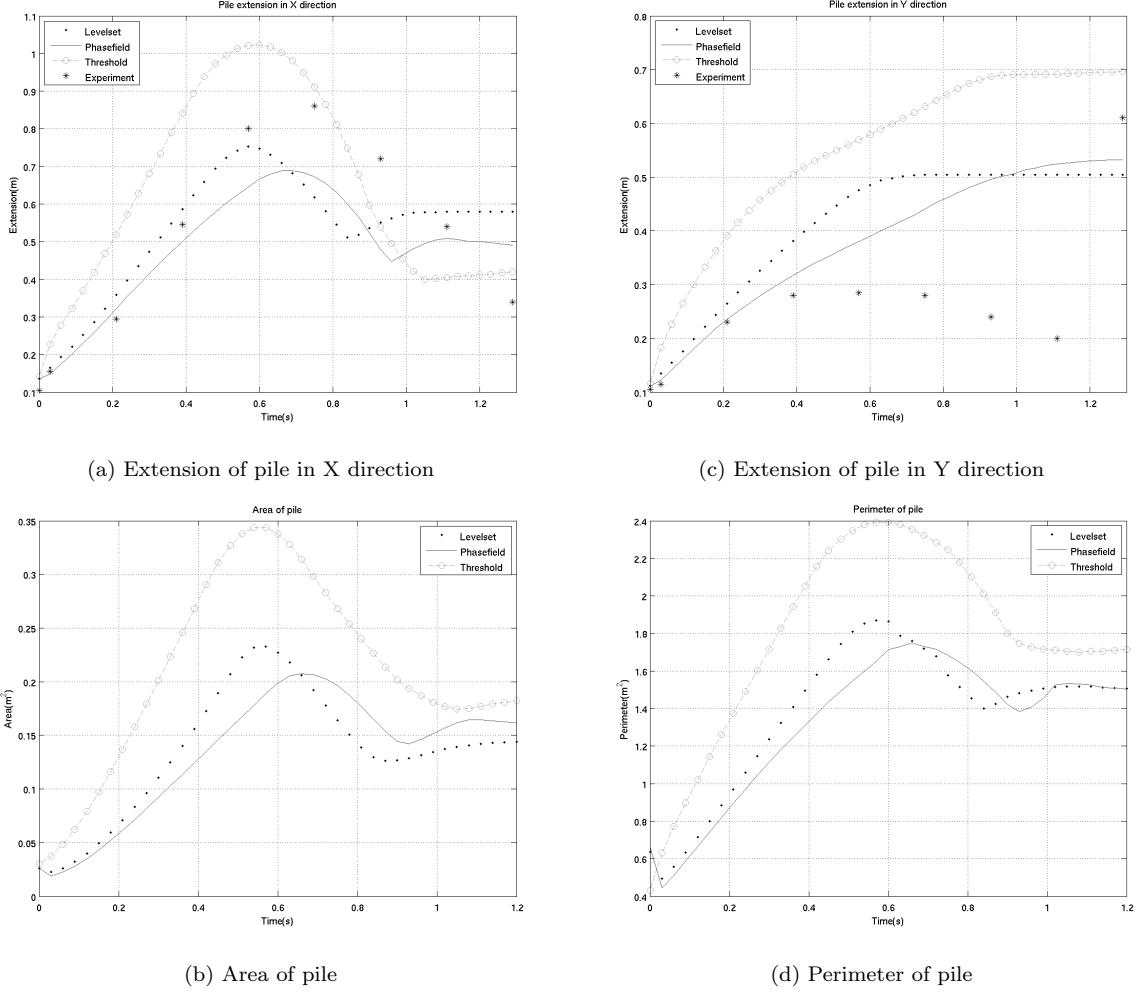


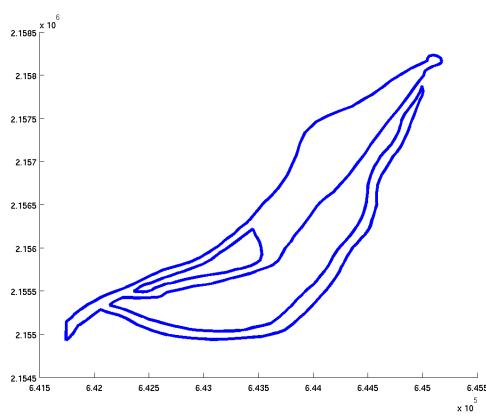
Figure 7: Comparison of the methods for flow on inclined plane

## 4.2 Colima Volcano

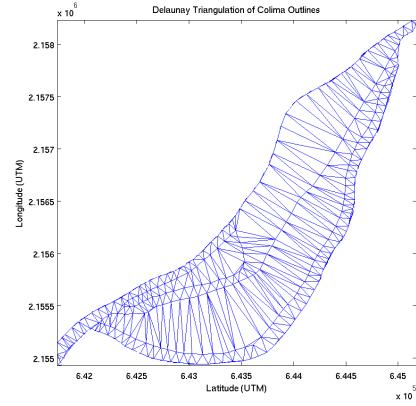
### keith wants the result of Atentique

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 volcanos in North America, and this eruption happened on 16-17 April 1991 [16]. 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  $h = h_{scale} \times GEOFLOW\_TINY$ , where  $h_{scale} = Volume^{\frac{1}{3}}$ .

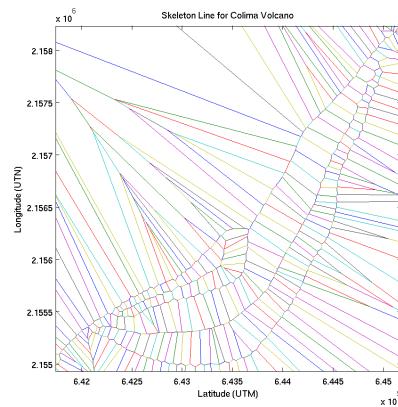
The resolution of the digital elevation model (DEM) of Colima volcano that here we used is 5 meter which



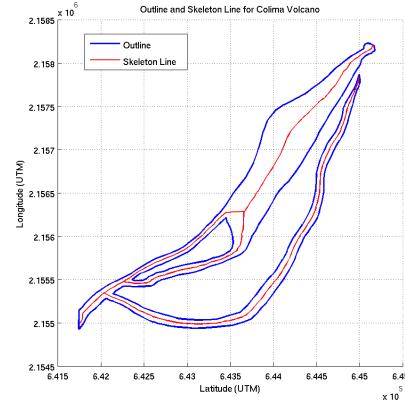
(a) Outline of flow from field data



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

is the finest DEM that we have ever used for this volcano.

The outline of this eruption is available in [36], we used the method that is described in [36] 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.

After finding the skeleton line, we mapped the outline of the flow and the skeleton line on the Colima volcano using Keyhole Markup Language (KML) and Google earth application 9. In figure 10 the result of each methods and their comparison can be seen.

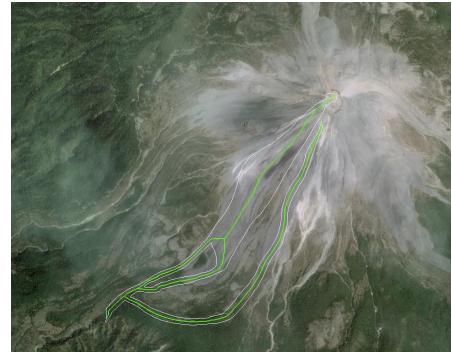
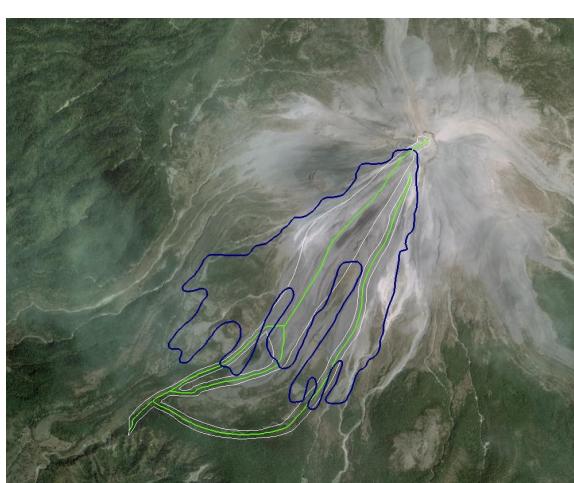
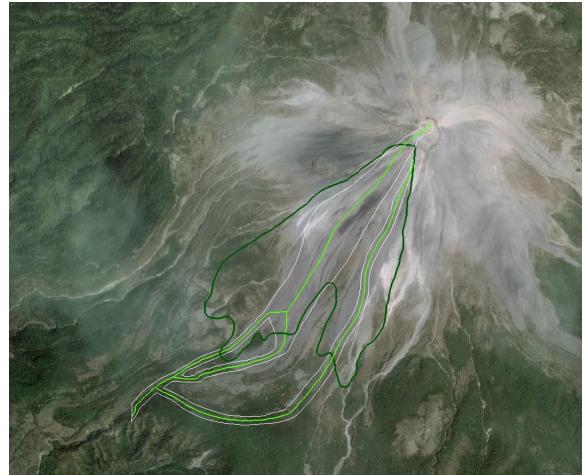


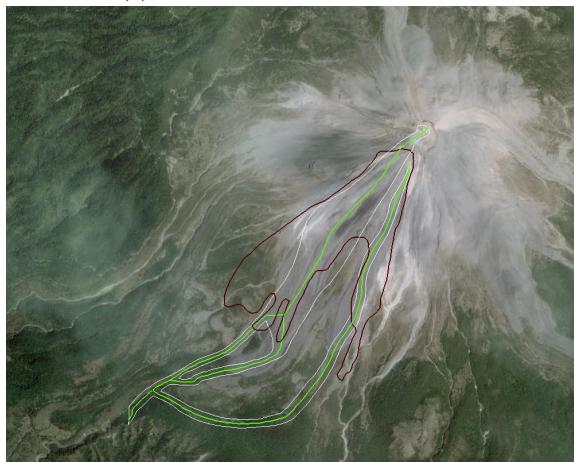
Figure 9: Skeleton line (white line) and outline of the deposit of eruption 1991 (green line)



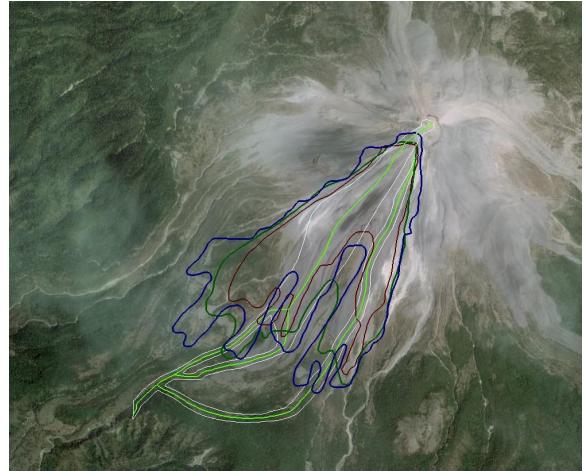
(a) Outline from Heuristic Method



(c) Outline from Phase field Method



(b) Outline from Level set Method



(d) Comparison of all three methods

Figure 10: Comparison of the obtained outline of flow from Heuristic, Level set, and Phase field methods for Colima Volcano

## 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.

## A Appendix 1

The constrain that we want to satisfy is to preserve the mass for entire flow. Since this flow is an incompressible flow, we can write:

$$\frac{D}{Dt} \int_{\Omega} \varphi h \, dx = 0. \quad (24)$$

with using Reynold's theorem, we can expand the above integral in to:

$$\int_{\Omega} \frac{\partial(\varphi h)}{\partial t} \, dx + \int_{\Gamma} (\vec{V} \cdot \vec{n})(\varphi h) \, ds = 0, \quad (25)$$

where  $\Gamma$  is the boundary of  $\Omega$ , and  $\vec{n}$  is its normal vector, and  $\vec{V}$  is velocity vector. The boundary and initial conditions for  $\varphi$  and velocity are:

$$\begin{aligned} \vec{V}_{t=0} &= \vec{V}_0(\mathbf{x}), \quad \varphi_{t=0} = \varphi_0(\mathbf{x}), \quad x \in \Omega \\ \frac{\partial \varphi}{\partial n} |_{\Gamma} &= 0. \end{aligned}$$

With the above initial and boundary conditions, and Gauss' identity, the equation (25) can be written:

$$\underbrace{\int_{\Omega} \varphi \frac{\partial h}{\partial t} dx}_a + \underbrace{\int_{\Omega} h \frac{\partial \varphi}{\partial t} dx}_b + \underbrace{\int_{\Omega} \varphi \nabla \cdot (h \vec{V}) dx}_c + \underbrace{\int_{\Omega} h \vec{V} \cdot \nabla \varphi dx}_d = 0. \quad (26)$$

We know  $a + c = 0$  from conservation of mass (equation (1)), so  $b + d = 0$ :

$$\int_{\Omega} h \left( \frac{\partial \varphi}{\partial t} + \vec{V} \cdot \nabla \varphi \right) dx = 0. \quad (27)$$

Since in the above equation, the expression inside the preanthesis is the LHS of phase field equation (11), we can substitute it with the RHS of the same equation, so:

$$\int_{\Omega} h \left( \frac{\partial \varphi}{\partial t} + \vec{V} \cdot \nabla \varphi \right) dx = \int_{\Omega} h \gamma (\Delta \varphi - F'(\varphi) + \xi(t)) dx = 0. \quad (28)$$

Applying the Gauss' theorem on laplacian term, with considering the boundary conditions on  $\varphi$ :

$$\int_{\Omega} h \gamma \nabla \cdot \nabla \varphi dx = \int_{\Gamma} h \gamma \nabla \varphi \cdot \vec{n} ds = 0, \quad (29)$$

consequently equation (28) leads to:

$$\xi(t) \int_{\Omega} h dx = \int_{\Gamma} h F'(\varphi) dx, \quad (30)$$

and we know intial volume is equal to  $\int_{\Omega} h dx$ , therefore we can conclude:

$$\xi(t) = \frac{1}{V_0} \int_{\Gamma} h F'(\varphi) dx. \quad (31)$$

## References

- [1] Operators on Inner-Product Spaces. pages 127–161.
- [2] D Adalsteinsson and J.a Sethian. The Fast Construction of Extension Velocities in Level Set Methods. *Journal of Computational Physics*, 148(1):2–22, January 1999.
- [3] David Adalsteinsson and James A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269 – 277, 1995.
- [4] a.K. Patra, a.C. Bauer, C.C. Nichita, E.B. Pitman, M.F. Sheridan, M. Bursik, B. Rupp, a. Webber, a.J. Stinton, L.M. Namikawa, and C.S. Renschler. Parallel adaptive numerical simulation of dry avalanches over natural terrain. *Journal of Volcanology and Geothermal Research*, 139(1-2):1–21, January 2005.
- [5] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30(1):139–165, 1998.
- [6] F. Aureli, a. Maranzoni, P. Mignosa, and C. Ziveri. A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Advances in Water Resources*, 31(7):962–974, July 2008.
- [7] Satish Balay, Shirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.5, Argonne National Laboratory, 2014.
- [8] Andrea Balzano. Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coastal Engineering*, pages 83–107, 1998.
- [9] W. J. Boettinger, J. a. Warren, C. Beckermann, and a. Karma. P Hase -F Ield S Imulation of S Olidification 1. *Annual Review of Materials Research*, 32(1):163–194, August 2002.
- [10] Lia Bronsard and Robert V. Kohn. On the slowness of phase boundary motion in one space dimension. *Communications on Pure and Applied Mathematics*, 43(8):983–997, 1990.
- [11] Shintaro Bunya, Ethan J. Kubatko, Joannes J. Westerink, and Clint Dawson. A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1548–1562, April 2009.
- [12] John W. Cahn. Free energy of a nonuniform system. ii. thermodynamic basis. *The Journal of Chemical Physics*, 30(5), 1959.
- [13] John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics*, 28(2), 1958.
- [14] M.J. Castro, a.M. Ferreiro Ferreiro, J.a. García-Rodríguez, J.M. González-Vida, J. Macías, C. Parés, and M. Elena Vázquez-Cendón. The numerical treatment of wet/dry fronts in shallow flows: application to one-layer and two-layer systems. *Mathematical and Computer Modelling*, 42(3-4):419–439, August 2005.
- [15] Vincenzo Casulli. Ahigh-resolution wetting and drying algorithm for free-surface hydrodynamics. *International journal for numerical methods in fluids*, (August 2008):391–408, 2009.
- [16] S.J. Charbonnier and R. Gertisser. Field observations and surface characteristics of pristine block-and-ash flow deposits from the 2006 eruption of merapi volcano, java, indonesia. *Journal of Volcanology and Geothermal Research*, 177(4):971 – 982, 2008. Volcanic Flows and Falls A Special Issue to Honor Michael F. Sheridan.

- [17] Long-Qing Chen. PHASE-FIELD MODELS FOR MICROSTRUCTURE EVOLUTION. *Annual Review of Materials Research*, 32(1):113–140, August 2002.
- [18] Liang Cheng and Steven Armfield. A simplified marker and cell method for unsteady flows on non-staggered grids. *International Journal for Numerical Methods in Fluids*, 21(1):15–34, 1995.
- [19] Keith R Dalbey. PREDICTIVE SIMULATION AND MODEL BASED HAZARD MAPS UMI Number : 3356020 Copyright 2009 by Dalbey , Keith R . All rights reserved. 2009.
- [20] L. D'Alpaos and a. Defina. Mathematical modeling of tidal hydrodynamics in shallow lagoons: A review of open issues and applications to the Venice lagoon. *Computers & Geosciences*, 33(4):476–496, May 2007.
- [21] M. E. Eglit and Ye I Sveshnikova. Matematicheskoye modelirovaniye snezhnykh lavin; mathematical modeling of snowavalanches. *Materialy Glyatsiologicheskikh Issledovaniy, Khronika Obsuzhdeniya*, 38:79–84, 1980.
- [22] L. Fraccarollo and E.F. Toro. Experimental and numerical assessment of the shallow water model for two dimensional dam-break type problems. *Journal of Hydraulic Research*, 33:843–864, 1995.
- [23] D. Gerlach, G. Tomar, G. Biswas, and F. Durst. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat and Mass Transfer*, 49(3-4):740–754, 2006.
- [24] James Glimm, John W. Grove, Xiao Lin Li, Keh-Ming Shyue, Yanni Zeng, and Qiang Zhang. Three dimensional front tracking. *SIAM J. Sci. Comp*, 19:703–727, 1995.
- [25] V.R. Gopala and B.G.M. van Wachem. Volume of fluid methods for immiscible-fluid and free-surface flows. *Chemical Engineering Journal*, 2008.
- [26] J. M. N. T. Gray, M. Wieland, and K. Hutter. Gravity-driven free surface flow of granular avalanches over complex basal topography. *Royal Society of London Proceedings Series A*, 455:1841–1874, May 1999.
- [27] Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.
- [28] C.W. HIRT and BD NICHOLS. Volume of fluid/VOF/ method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.
- [29] R. M. Iverson. The physics of debris flows. *Reviews of Geophysics*, 35:245–296, 1997.
- [30] R. M. Iverson and R. P. Denlinger. Flow of variably fluidized granular masses across three-dimensional terrain 1. Coulomb mixture theory. *Journal of Geophysical Research*, 106:537–552, 2001.
- [31] Georges Kesserwani and Qiuhua Liang. Locally Limited and Fully Conserved RKDG2 Shallow Water Solutions with Wetting and Drying. *Journal of Scientific Computing*, 50(1):120–144, March 2011.
- [32] K.Hutter, M. Siegel, S.B. Savage, and Y. Nohguchi. Two-dimensional spreading of a granular avalanche down an inclined plane part i. theory. *Acta Mechanica*, 100:37–68, 1993.
- [33] Junseok Kim. Phase-Field Models for Multi-Component Fluid Flows. *Commun. Comput. Phys*, 12(3):613–661, 2012.
- [34] R.G. Larson. *The Structure and Rheology of Complex Fluids*. Topics in Chemical Engineering. OUP USA, 1999.

- [35] SC Medeiros and SC Hagen. Review of wetting and drying algorithms for numerical tidal flow models. *International Journal for Numerical ...*, (March 2012):473–487, 2013.
- [36] Laercio M. Namikawa. *Multiple Representations of Elevation for Dynamic Process Modeling*. PhD thesis, Department of Geography, University at Buffalo, July 2006.
- [37] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.
- [38] A.K. Patra, A.C. Bauer, C.C. Nichita, E.B. Pitman, M.F. Sheridan, M.I. Bursik, B. Rupp, A. Webber, A.J. Stinton, L.M. Namikawa, and C.S. Renschler. Parallel adaptive numerical simulation of dry avalanches over natural terrain. *Journal of Volcanology and Geothermal Research*, 139:1–21, 2005.
- [39] A.K. Patra, C.C. Nichita, A.C. Bauer, E.B. Pitman, M.I. Bursik, and M.F. Sheridan. Parallel adaptive discontinuous galerkin approximation for thin layer avalanche modeling. *Computers and Geosciences*, 32:912–926, 2006.
- [40] E.B. Pitman, A. Patra, A. Bauer, C. Nichita, M. Sheridan, and M.Bursik. Computing debris flows. *Physics of Fluids*, 15:3638–3646, 2003.
- [41] S. P. Pudasaini and K. Hutter. Rapid shear flows of dry granular masses down curved and twisted channels. *Journal of Fluid Mechanics*, 495:193–208, November 2003.
- [42] M. Quecedo and M. Pastor. A reappraisal of Taylor-Galerkin algorithm for drying-wetting areas in shallow water computations. *International Journal for Numerical Methods in Fluids*, 38(6):515–531, 2002.
- [43] S.B. Savage and K. Hutter. The motion of a finite mass of granular material down a rough incline. *Journal of Fluid Mechanics*, 199:177–215, 1989.
- [44] SB Savage and RM Iverson. Surge dynamics coupled to pore-pressure evolution in debris flows. *Debris-Flow Hazards Mitigation: Mechanics, Prediction and Assessment*, edited by: Rickenmann, D. and Chen, C.-L., Millpress, 1:503–514, 2003.
- [45] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146 – 159, 1994.
- [46] Y. C. Tai, S. Noelle, J. M. N. T. Gray, and K. Hutter. Shock-capturing and front-tracking methods for granular avalanches. *Journal of Computational Physics*, 175(1):269–301, 2002.
- [47] E.F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows*. Wiley, New York, 2001.
- [48] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 2009.
- [49] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100(1):25–37, May 1992.
- [50] Luiz C Wrobel and CA Brebbia. *Computational Modelling of Free and Moving Boundary Problems: Fluid flow*, volume 1. Walter de Gruyter, 1991.
- [51] Xiaofeng Yang, James J. Feng, Chun Liu, and Jie Shen. Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method. *Journal of Computational Physics*, 218(1):417 – 428, 2006.
- [52] DL Youngs. Time-dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285. Academic Press, 1982.