Benchmark Validation and Bayesian Analysis of Lava Flow Model Performance

Jacob Richardson

December 5, 2015

Abstract

Modeling lava flows through cellular automata (CA) methods enables a computationally inexpensive means to quickly forecast lava flow paths and ultimate areal extents. A CA program has been created in the program language C that is modular, which enables a combination of governing CA rules to be evaluated against each other. My objective is to find a successful combination of automata behaviors that accurately forecasts lava inundation and behaves like a bingham fluid. To fulfill this objective, four benchmarking levels have been devised to test lava spreading algorithms against increasingly complex tests. These levels are 1) verification of the code by testing for conservation of mass; 2) testing for flow self-similarity given inconsequential variations of arbitrary surfaces; 3) testing for replication of Bingham flow morphology on simple surfaces and; 4) testing for replication of real lava flow morphologies on pre-eruptive surface models. Currently the best-fitting lava spreading algorithm for these four benchmarking tests might be an algorithm which spreads lava proportional to slope and where each automaton is able to spread lava in 8 directions.

1 Introduction

Lava flows as a gravity current on the surface of the Earth when liquid magma is effused at the surface with little or no explosivity. In the vacinity of active volcanoes, lava flows represent significant long term impact to infrastructure. In the past, lava flow hazard has been mitigated with the construction of physical diversions and at least once in 252 AD by the supernatural grace of St. Agatha of Sicily who died the year prior. Modern science suggests, however, that forecasting the flow path of lava from active volcanoes might be more useful than St. Agatha for communities impacted by effusive volcanism.

Methods of forecasting lava flows range from simple predictions using empirical relationships between magma flux and flow length [Glaze and Baloga, 2003], to 1-D numerical solutions such as FLOWGO [Harris and Rowland, 2001], to advanced computational fluid dynamics codes like lavaSIM [Hidaka et al., 2005]. All modern numerical flow models by nature trade precision in simulating physical processes with computer run-time, so that while FLOWGO is relatively fast it only predicts downslope flow length, while lavaSIM

solves Navier-Stokes equations to produce a 3-D flow distribution at the expense of large computational requirements.

Cellular Automata (CA) methods have been developed to simulate fluid flow, including lava spreading [Barca et al., 1994]. In contrast to CFD codes, these do not generally attempt to compute Navier-Stokes equations but instead abstract many physical parameters, such as viscosity and temperature, into more or less empirical rules. The benefit of CA methods for simulating lava flows is most noticeable in the reduced computer time necessary for simulation compared to CFD methods.

Multiple CA lava flow algorithms exist, such as SCIARA [Crisci et al., 2004], MAGFLOW [Del Negro et al., 2008], ELFM [Damiani et al., 2006], and LavaPL [Connor et al., 2012]. These algorithms are variations on a theme, where the largest difference between each is how lava is distributed from one automaton to its neighbors. For instance, three versions of SCIARA allow for lava to spread in cardinal directions [Barca et al., 1994], in hexagonal directions [Crisci et al., 2008], or in directions based on an inherent velocity calculated in an eulerian way for each automaton [Avolio et al., 2006]. MAGFLOW and ELFM by contrast to the original SCIARA algorithm implements 8 directions of spreading. LavaPL and SCIARA both spread in four directions but the apportionment of lava from one automaton to neighbors is based on a different algorithm.

While several lava flow simulators now exist, each have been made and tested with different lava flows or aspects of flows in mind. Because of this, selecting a specific algorithm to effectively model lava flow hazards can be a necessary, if unwanted challenge. To address this problem, we propose a hierarchical benchmarking scheme to objectively rank different flow spreading algorithms. This hierarchy tests simulated output against increasingly complex tests, from simply conserving mass to replicating the paths and ultimate areal extents of real lava flows. These benchmarking methods can be applied to any flow algorithm that provides at least a list or map of inundated locations over various topographies.

In this paper, multiple lava flow algorithms are tested using a new modular lava flow code, which I have named MOLASSES (standing for *MOdular LAva Simulation Software in Earth Science*). This code, implemented in C, is a Cellular Automata code which tracks a population of equal-area spaced cells over a grid, that is defined by a digital elevation model (DEM). These cells may or may not be inundated with lava and they are governed by universal rules. Because MOLASSES has been designed in a modular way, it is relatively quick to modify the flow algorithm. Using this code while changing methods of lava distribution enables code output in a constant format, which simplifies the comparison of methods.

In Section 2, I will demonstrate how CA is applied to lava flows and detail how a CA simulation is carried out in the MOLASSES code. I will introduce a hierarchy of benchmarks in Section 3 that can be used to verify and validate different lava flow algorithms using increasingly complex model parameters. In Section 4, I will expand on the final benchmark level (validation against real flows) with a Bayesian approach to improving model performance for the 2012-3 Tolbachik Lava Flow. The results from these Sections will be discussed in Section 5.

1.1 Case Study Area: 2012-3 Tolbachik Lava Flow

The Tolbachik lava flow began in November 2012, originally being sourced from a long fissure vent south of Tolbachik Dol. Initial magma flux was estimated to be 440 m^3 s^{-1} [Belousov et al., 2015]. The fissure vent ultimately coalesced into two main vents, seen in TanDEM-X data and the flux dropped significantly to between 100 and 200 m^3 s^{-1}. Early stages of the flow carried lava west to a maximum runout of 14.5 km and later stages beginning in January or February, carried lava east. The total emplacement volume is ∼0.5 cu. km. with 0.38 cu. km. of that being to the west. TanDEM-X data show that the modal thickness of the flow is 7.8 m, and that the overall thickness distribution is log-normal. After the flow ceased, the total emplacement area was mapped using orthophotos and TanDEM-X data where clouds were present in the images by Kubanek et al. [2015].

2 A Modular Cellular Automata Algorithm for lava flows

CA in lava flows has historically been defined as a 2-dimensional space, which is divided into equal-area grid cells, such as those found in a common digital elevation model (DEM). Within the location of each cell is defined an "elementary automaton" (*ea*) that has a set of properties, is governed by a set of global rules, and has a set list of neighboring automata. While the behavior rules that each *ea* follow is identical to those of all other automata, its behavior is only dictated by local phenomena. Specifically, the amount of lava that flows in or out of an *ea* will depend on properties such as lava thickness and elevation within it and its neighbors. Because grid cells and *ea* are fundamentally inseperable in this application, I will refer to ea as cells.

The set of cellular automata is defined as

$$\mathbf{A} = \{\mathrm{E}^2, \mathrm{V}, \mathrm{S}, \mathrm{X}, \sigma, \gamma\} \tag{1}$$

where E^2 is the set of point locations of cells in $\mathbf{A}$, $\mathrm{V} \subset \mathrm{E}^2$ is the set of vent or source locations, S is the set of substates within each cell, and X is the local neighborhood that each cell can directly influence [Barca et al., 1994]. σ and γ represent the transition functions and source functions within $\mathbf{A}$.

Practically, E^2 is a set of coordinate pairs denoting row and column addresses of cells in a larger grid. $\mathrm{S}(i,j)$, which represents the set of substates for the cell at row i, column j, includes S_e, the underlying elevation of an automaton; S_h, the thickness of lava within the cell; and S_{h0}, the critical thickness, above which lava will spread from a cell. Some algorithms include S_T, or the cell temperature in this set. X, in a four-connected neighborhood scheme, is given as {(0,1), (0,-1), (1,0), (-1,0)}, where (0,0) is the location of a cell under evaluation. σ is the change of substates in S for each cell from timestep t to $t+1$, or $\mathrm{S}^t \rightarrow \mathrm{S}^{t+1}$. γ specifies the lava emitted at locations within V. The implementation of these sets within the CA structure $\mathbf{A}$ is described in detail below.

2.1 MOLASSES Algorithm Outline

MOLASSES is a Cellular Automata code developed in the C programming language based on the CA algorithm "LavaPL" of Connor et al. [2012]. The major change between LavaPL

and MOLASSES is that MOLASSES is constructed with nine modules that each have a specific task, either carrying out the CA simulation, reading model input, or writing model output (Figure 1). The nine modules were designed to replicated major functions in LavaPL and are:

1. **DRIVER** Calls modules in sequence to execute the flow algorithm.
2. **INITIALIZE** Reads a user-provided configuration file to define model parameters.
3. **DEM_LOADER** Imports a raster file to define the elevation model.
4. **INITFLOW** Uses model parameters to define data arrays.
5. **PULSE** Incrementally adds lava to source locations.
6. **DISTRIBUTE** Determines whether to spread and how to spread lava between cells.
7. **NEIGHBOR_ID** Identifies the cell neighborhood.
8. **ACTIVATE** Adds newly inundated cells to the list of active cells.
9. **OUTPUT** Writes model results to a file using user-specified formats.

Like LavaPL, model parameters are specified by a user through a text configuration file, which must include 1) a digital elevation model (DEM), 2) a residual lava flow thickness, 3) at least one vent location, 4) the total volume and "pulse volume" of this vent, and 5) an output file path. The lava flow thickness defines the CA value of S_{h0}, where cells with flow thicknesses $S_h > S_{h0}$ will spread all lava to their neighboring cells, while cells with less lava will retain their lava. The "pulse volume" defines γ and the amount of lava to emit at the source location at each time step. The total volume constrains γ as lava will not be introduced to the source location after the total volume has been delivered. Modules within MOLASSES that further execute the CA simulation are detailed below.

2.2 Cells in E^2

Information for cells in the grid defined by E^2 is stored in two ways, for code efficiency. First, some information of the CA structure **A** is stored in a Global Data Grid. This grid stores information known at the beginning of the simulation, such as the user supplied residual flow thickness and the elevation. Grid dimensions are set in the **DEM_LOADER** module to be identical to the user-specified raster DEM. This module then imports the elevation of each raster pixel into the corresponding grid cell location. After this operation, the residual flow thickness is also stored in the grid.

The second information storage method is a list defined in the **INITFLOW** module. The "Active List" is declared with a length that corresponds to the theoretical maximum number of cells that can be inundated by lava. This list contains data that is updated during the simulation, including lava thicknesses, S_h, within cells. As cells are determined within the simulation to be inundated with lava for the first time, their row and column addresses, as well as their lava thicknesses are appended to the Active List with the module **ACTIVATE**.

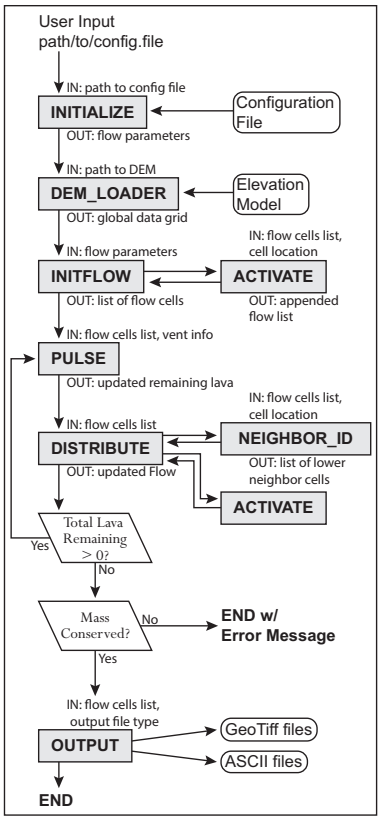

Figure 1: A flow chart of MOLASSES carried out within the **DRIVER** module. Gray boxes denote various modules, with major inputs and outputs given above and below. Parallelograms are checks performed within DRIVER itself. Rounded boxes represent external input and output files.

2.3 Source Locations, V, and the Source Function, γ

Initially in the Active List, **INITFLOW** only declares source location(s) as the first few elements of the list. These source locations are flagged in the list to be identified as source locations by other modules.

The **PULSE** module carries out the source function, γ. In this module, a separate array stores each source vent's volume parameters. The pulse volume is added to the quantity of lava in the source cell and is subtracted from the remaining volume. The remaining volume is initially set as the total volume given in the configuration file, so PULSE continues to add lava to the source locations at each time step until remaining volume is 0.

2.4 Substates, S, and the Transition Function, σ

Substates which cannot change, such as the cell elevation S_e and the residual flow thickness S_{h0}, are stored within the Global Data Grid. Substates which do change, primarily flow thickness, S_h, are stored in the Active List and are allowed to change from timestep to timestep. These values are initialized in **INITFLOW** where thicknesses are set to 0.

The transition function, σ, is defined in the **DISTRIBUTE** module. Cells in this module are evaluated in order of their inundation (i.e. vents are evaluated first and distal cells are

evaluated last). The incoming and outgoing quantity of lava from each cell is stored in the Active List. Generally, if a cell has a flow thicknesses $S_h > S_{h0}$, it will spread the lava above S_{h0} to any neighbors lower in elevation than itself. When all inundated cells have been evaluated, the incoming and outgoing quantities of lava of each cell are applied to the cells. This flow transition represents a timestep as all cells are updated at once.

Multiple possible transition functions can effectively spread lava from and to cells in a manner that might replicate lava in real life. Selecting the best transition function is the purpose of the validation benchmarks described in Section 3. In this project three main variations are combined and tested which vary 1) how local slope affects spreading, 2) the neighborhood size, and 3) if any neighbors are eliminated from the neighborhood based on their relationship to the cell.

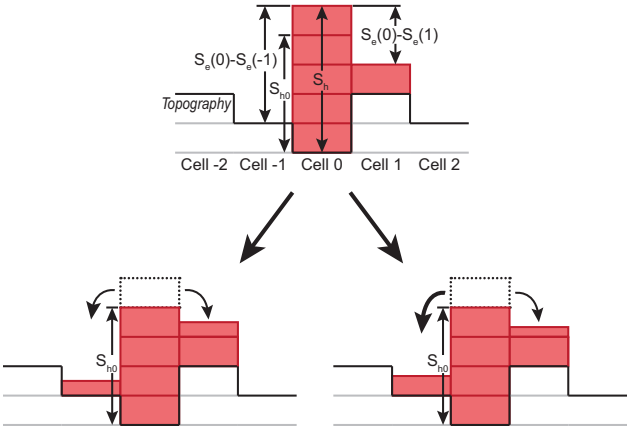

Figure 2: A 2-D example of two transition functions. At timestep t (top), Two cells are inundated with lava. The central cell (Cell 0) has 1 block of lava higher than the residual thickness, S_{h0}. In a slope-proportional sharing scheme, timestep $t+1$ will follow the path to the right; because Cell -1 has twice the relief as Cell 1, it receives twice as much of the residual lava (2/3 blocks vs. 1/3 to the right). In an equal-sharing scheme, the left path will be followed, and half the block will be added to both neighbor cells.

Local slope-based spreading In the LavaPL algorithm given by Connor et al. [2012], lava is apportioned from cells to their neighboring cells proportional to slope. To give a specific case, let a cell at location c be the central cell, with a set of neighbor cells, X. The total relief between cell c and its lower neighboring cells is

$$\mathrm{TR}(c) = \sum^{n \in \mathrm{X}} (\mathrm{S}_h(c) + \mathrm{S}_e(c)) - (\mathrm{S}_h(n) + \mathrm{S}_e(n)) \tag{2}$$

where S_h is the height or thickness of the lava in a cell, S_e is the underlying elevation of the cell, and n is a neighbor in X. The total lava to spread away from the central cell is the difference between thickness of lava (S_h) at c the residual thickness (S_{h0}), unless the lava thickness is lower than the residual thickness, giving

$$\mathrm{Outbound}(c) = \begin{cases} \mathrm{S}_h(c) - \mathrm{S}_{h0}(c) & \text{if } S_h(c) - \mathrm{S}_{h0}(c) > 0 \\ 0 & \text{if } S_h(c) - \mathrm{S}_{h0}(c) \leq 0 \end{cases}$$

In LavaPL, the excess flow, "Outbound", is delivered to neighbors n based on the proportion of total relief, TR, found at each neighbor location (the right path of Figure 2). For each $n \in$X,

$$\text{Inbound}(n) = \text{Outbound}(c) \left(\frac{(\text{S}_h(c) + \text{S}_e(c)) - (\text{S}_h(n) + \text{S}_e(n))}{\text{TR}} \right) \tag{4}$$

This is the slope-proportional spreading equation. Another method would be "slope-blind," and would spread lava to all lower neighbors equally following the equation

$$\text{Inbound}(n) = \left(\frac{\text{Outbound}(c)}{|\text{X}|} \right) \tag{5}$$

where |X| is the size of the neighborhood, or the number of elements in the neighborhood. This is illustrated as the left path of Figure 2.

Neighborhood size The size of the neighborhood, X, in CA algorithms is commonly 4 or 8 in cardinal or ordinal directions. Here both have been implemented, which enables the benchmarks to test whether 8 spreading directions increases the performance of these tests. This is further described in the next section (2.5).

Spreading inhibited by special relationships Though the size of the neighborhood is set globally for all cells, neighbors are not guaranteed to receive lava from central cells. In all algorithms, for example, cells in the neighborhood that are higher than the central cell, including lava thicknesses, are excluded from the neighborhood set.

Other neighbor elimination rules can also be implemented. One has been designed by Connor et al. [2012], where the cell that initially gives lava to another cell is forever eliminated from the receiving cell's neighborhood. This is done by creating a "parent-child" relationship for each activated cell in the flow. Simply put, child cells cannot give lava to their parent cells (right path in Figure 3). This transition function rule is tested against no parentage rules in competing MOLASSES algorithms (left path in Figure 3).

2.5 Cell Neighborhood, X

The final set in the CA is the cell neighborhood X and is defined by the **NEIGHBOR_ID** module. This neighborhood is usually either 4-connected (von Neumann neighborhood) or 8-connected (Moore neighborhood) as illustrated in Figure 4. Four-connected neighborhoods are defined as the row, column coordinates {(0,1), (0,-1), (1,0), (-1,0)}, where (0,0) is the location of a cell under evaluation, while the set elements might correspond to North, South, East, and West. Eight-connected neighbors include the ordinal directions, Northeast, Southeast, Northwest, and Southwest: {(0,1), (0,-1), (1,0), (-1,0), (1,1), (-1,-1), (1,-1), (-1,-1)}.

NEIGHBOR_ID is implemented within the DISTRIBUTE module to evaluate cells within X, and determine whether they are lower in elevation (including their lava) than the central

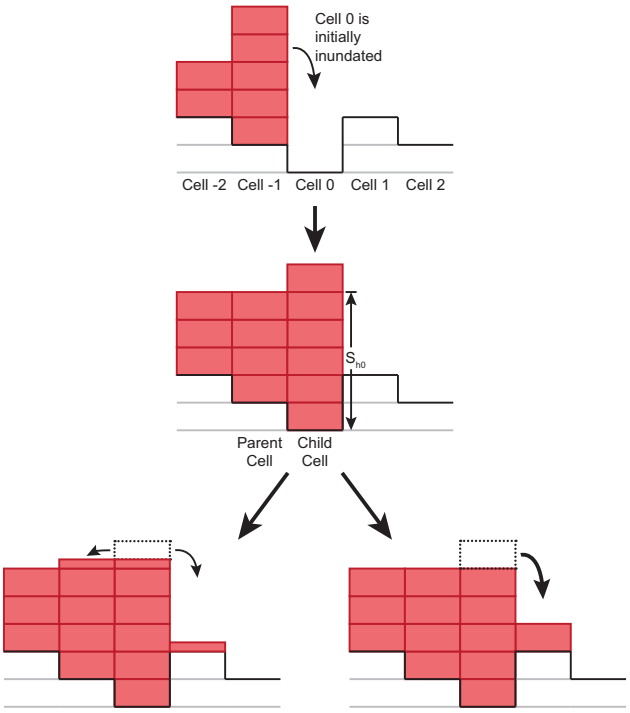

Figure 3: Another 2-D example of different transition functions. In the first timestep (top), Cell -1 initially inundates Cell 0, creating the Parent-Child relationship shown in the next illustrated timestep (middle). If Parents cannot receive lava from Child cells, all residual lava in Cell 0 will flow to Cell 1, following the path to the right. If these relationships are ignored, as shown in the left path, Cell 0 will spread lava in both directions.

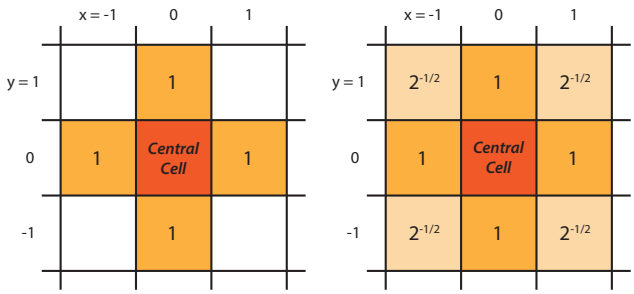

Figure 4: Cellular Automata neighborhoods. To the left, in a 4-connected neighborhood, a central cell may influence or be influenced by cells in cardinal directions. To the right, in an 8-connected neighborhood, the zone of influence is expanded to include ordinal directions. Numbers in each cell are relative weights (determined by distance from the central cell), so diagonal neighbors are weighted less than orthogonal cells.

cell. If one is lower, NEIGHBOR_ID returns their relief, or the difference in elevation between the cell and the central cell, to the DISTRIBUTE module. Depending on whether parent-child relationships are recorded or ignored in the transition function, NEIGHBOR_ID can follow one of two algorithms below.

4-connected NEIGHBOR_ID module	**8-connected NEIGHBOR_ID with Parent-Child Relationships**
X = {(0,1), (0,-1), (1,0), (-1,0)}	X = {(0,1), (0,-1), (1,0), (-1,0), (1,1), (-1,-1), (1,-1), (-1,-1)}
X' = {}	X' = {}
c = (0,0) (central cell location)	c = (0,0) (central cell location)
For $n \in$ X	**For** $n \in$ X
If $(S_h(c) + S_e(c)) - (S_h(n) + S_e(n)) > 0$	**If** $(S_h(c) + S_e(c)) - (S_h(n) + S_e(n)) > 0$
Append n to X'	**If** n is **not** Parent of c
	Append n to X'
Return X'	**Return** X'

3 Benchmarking Hierarchy

The strategy implemented in this paper follows the advice of Bayarri et al. [2007] for validating computer models, namely "1) defining the problem; 2) establishing evaluation criteria; 3) designing experiments; 4) approximating computer model output; 5) analyzing the combination of field and computer run data." The sixth step in their validation process, feeding results back to revise models, has been done informally to determine how to alter spreading algorithms in future benchmarking attempts. Each level below presents a problem for a lava spreading algorithm to complete. These fundamental problems (e.g. replicating a Bingham flow) are evaluated using simple tests that demonstrate the problem. The relevant model output for each of these tests is a list of locations (i.e. a list of x and y coordinates) that have been inundated by lava. After verification (Level 0), the first validation level tests model results with other model results; the second level tests model output against expected analytical solutions; and the third level tests model output from field data.

Table 1: Transition Algorithm Codes and Descriptions

Transition Function	Neighborhood	Parent-Child Relationships Preserved?	Slope-proportional Sharing?
4/P/S	4-directions	Yes, "parents" do not accept lava from "children."	Yes, lower cells receive lava based on relative relief.
8/P/S	8-directions	Yes	Yes
4/N/S	4-directions	No, "parents" are not defined.	Yes
8/N/S	8-directions	No	Yes
4/P/E	4-directions	Yes	No, all lower cells receive equal quantities of lava.
8/P/E	8-directions	Yes	No
4/N/E	4-directions	No	No
8/N/E	8-directions	No	No

Test Algorithms Combining three variations of the Transition Function described in Section 2, I have created eight MOLASSES lava flow algorithms. Each variation has been made by modifying one module in the MOLASSES framework: The neighborhood is changed between 4- and 8- directions using the NEIGHBOR_ID module, classifying one cell as a "parent" cell when a location is initially inundated is within the ACTIVATE module, and dividing lava amongst neighboring cells proportional to slope or equally is carried out in the DISTRIBUTE module. These eight algorithms will be referred to using three character codes, listed in Table 1. For the algorithm used by LavaPL in Connor et al. [2012], the code would therefore be 4/P/S, as it spreads lava in 4-directions from a central cell, all inundated cells have designated parents to whom they cannot spread lava, and the quantity of lava to spread from a central cell is higher for lower neighboring cells.

3.1 Level 0: Conservation of Mass

Before the results of a lava flow simulation can be validated, it must be verified to at least prove that conservation of mass is preserved. A lava flow simulation will therefore not be tested against the following benchmark tests until this conservation of mass requirement is shown to be fulfilled.

In MOLASSES, the code is verified within the DRIVER module, which manages each subordinate module. The erupted volume, V_{in}, is given as the total eruption volume specified by the user in the configuration file. If multiple source locations are given in this file, V_{in} is the sum of total eruption volumes. V_{in} is compared at the end of the module to the total volume of the flow, or V_{out}. The volume V_{out} is calculated by summing the volume in all inundated grid cells. MOLASSES reports success if $V_{in} - V_{out} \leq 10^{-8}$ m^3, which is the precision of a 64-bit double. If this test fails, MOLASSES reports failure and the excess volume found in the flow.

MOLASSES Conservation of Mass Test

If $|V_{in} - V_{out}| \leq 10^{-8}$
  Print `SUCCESS: MASS CONSERVED`
Else
  $excess = V_{out} - V_{in}$
  Print `ERROR: MASS NOT CONSERVED! Excess:` $excess$ `m^3`

3.2 Level 1: Repeatability given meaningless parameter variation

Once the code has been verified to conserve mass, the flow can be validated. This first validation level tests that lava flow simulations are repeatable, regardless of changes in parameter space that should have no effect on the flow. Parameters that ideally should not effect lava flows include slope direction and elevation model resolution. For instance, a slope to the west and an identically dipping slope to the east should produce lava flows of equal length and shape (given identical flow attributes).

Miyamoto and Sasaki [1997] performed a simple validation test on two CA-like flow simulators [Ishihara et al., 1990, Miyamoto and Sasaki, 1997] where a sloped DEM was rotated 45 degrees from "south" to "southeast". This benchmark was performed to demonstrate that the flow models had the same run-out length regardless of the arbitrary slope direction. Here, the DEM rotation scheme by Miyamoto and Sasaki [1997] is adopted and expanded, so that a DEM of a simple slope is rotated 19 times at increments of 5°. Flows are simulated on each of these slopes and the locations of inundated cells are output from the model.

Three characteristics of the simulated flows are determined for each slope direction: flow length, orientation, and aspect ratio. Flow length is defined as the distance between the vent and the furthest inundated point from the vent. Flow orientation is defined as the direction that furthest point lies, with respect to North. Flow aspect ratio is the ratio of maximum flow width to flow length. Perfect success for this benchmark is when simulated flows, regardless of slope direction, 1) do not change in length, 2) have an orientation identical to the slope direction, and 3) do not change in aspect ratio. Failure is more subjective, but I will define failure as 1) more than 10% variation in flow length depending on slope direction, 2) more than 5° offset between the slope and the flow orientation on average, or 3) more than 15% variation in flow aspect ratio.

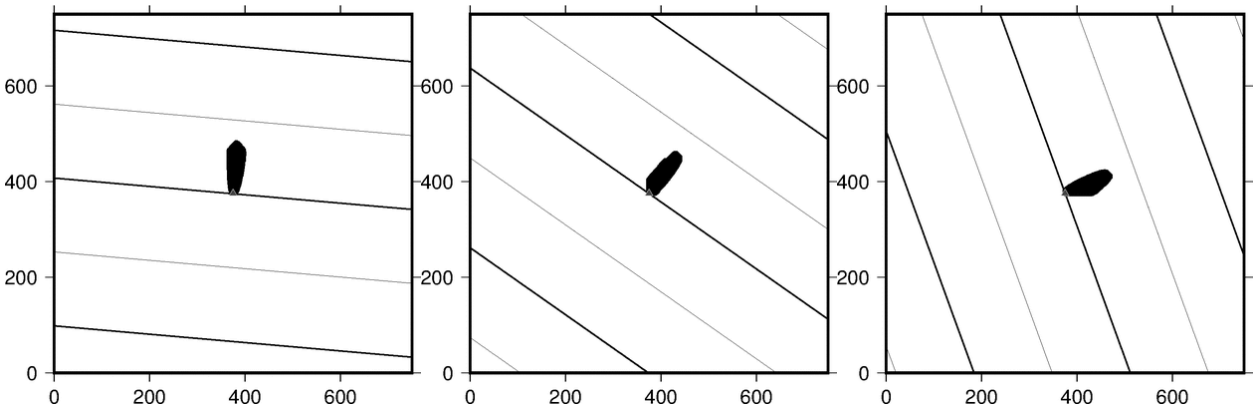

Figure 5: Rotating slope test for algorithm **4/P/S** (LavaPL). Slope dip is 18°, with dip-directions 0N, 30N, and 80N from left to right. The flow length and aspect ratio are similar and the flow direction is in the slope direction, so it passes Level 1 criteria.

3.2.1 Benchmark Parameters

The underlying DEM for this benchmark has a simple 18° slope, dipping to the North. The DEM has a spatial resolution of 1 m. The source cell is placed at the center of the DEM, is given a total volume of 1000 m^3, and is given a pulse volume of 1 m^3. When the simulation is finished, model output is used to determing the three flow characteristics used in this benchmark (length, orientation, and aspect ratio). The DEM is rotated 5° clockwise and the process is repeated 19 times until the flow is simulated on an East-facing slope.

3.2.2 Results

For all eight flow algorithms, flow length, aspect ratio, and orientation were calculated 19 times, corresponding to the 19 dip directions sampled between 0°N and 90°N. Variance for length and aspect ratio were calculated as the ratio of their standard deviations to their means. For instance, if mean runout length for the 18 flows is 100 m and the standard deviation of the 18 lengths is 2 m, the runout length variance is 2%. The mean direction error is also calculated for the set of flows from each algorithm. These are reported in the table below.

DEM Rotation Results

Transition Function	Run-out Variance	Aspect Ratio Variance	Mean Direction Error
4/P/S	2.7%	6.7%	1.2°
8/P/S	4.4	12.2	0.9
4/N/S	9.6	19.7	1.3
8/N/S	3.9	7.5	0.6
4/P/E	21.6	38.6	14.2
8/P/E	7.2	13.8	5.4
4/N/E	21.6	38.7	14.1
8/N/E	7.2	13.8	5.5

While with an ideal spreading algorithm, variances and direction error would be 0 under a rotating slope, every spreading algorithm tested performed differently as DEM direction changed. Following from the above pass-fail standards, five of the eight algorithms can be rejected. Algorithms 4/P/E and 4/N/E have high run-out length variance. Algorithms 4/N/S, 4/P/E and 4/N/E have large aspect-ratio variance. Algorithms 4/P/E, 8/P/E, 4/N/E, and 8/N/E all systematically deviate from running downslope by $> 5°$ on average. This implies that algorithms which share lavas equally from central cells to all lower neighboring cells perform worse than algorithms which share lavas proportional to slope.

For the eight different transition functions tested, runout length varied between 60-160 m. The flow algorithm with the least flow length variance was the 4-connected, parent-child, slope-proportional strategy implemented in LavaPL. Algorithms 4/P/S (LavaPL), 8/P/S, and 8/N/S cannot be rejected because of any of the three standards set in this benchmark.

3.3 Level 2: Replication of flow morphologies on simple physical surfaces

The second benchmarking level is the first step in validating lava flow algorithms against realistic flow expectations. Instead of parameter space being arbitrarily defined, which was the case in Level 1, the defined parameter space informs tests at this level as to what the model output should be. As lava flows on a large scale are well described as Bingham fluids, simulations can be tested against analytical solutions or experimental observations of these fluids in simple conditions. For instance, a lava flow on a perfectly flat surface might be expected to create a circular areal extent [Griffiths, 2000].

Here I measure flow algorithm performance on a flat surface from a single vent source location. To measure the extent to which the simulated flow replicates a circle, the inundated

area is compared to the area of a circle which circumscribes the flow exactly. This can be described as

$$Fit = \frac{A_{flow}}{\pi d_{max}^2} \tag{6}$$

where d_{max} is the farthest extent of the simulated flow from the vent. A perfect match to a circle would result in a $Fit = 1$. With the same maximum distance from the vent (i.e. the distance from the center to a vertex) a perfect square would cover 64% of the area of a circle, ergo $Fit = 0.64$. An octagon would have a fit of 0.90. We consider a model to successfully pass this test if it produces a flow of $Fit > 0.90$, or if the flow approximates a circle better than an octagon. The model unambiguously fails this test if if produces a flow of $Fit < 0.64$, where a square better describes a circle than a flow generated from the model.

Circular Flow Test

Fit = 1.0	Best Possible Score; perfectly circular.
Fit > 0.90	Success; better than an octagon.
Fit < 0.64	Failure; worse than a square.

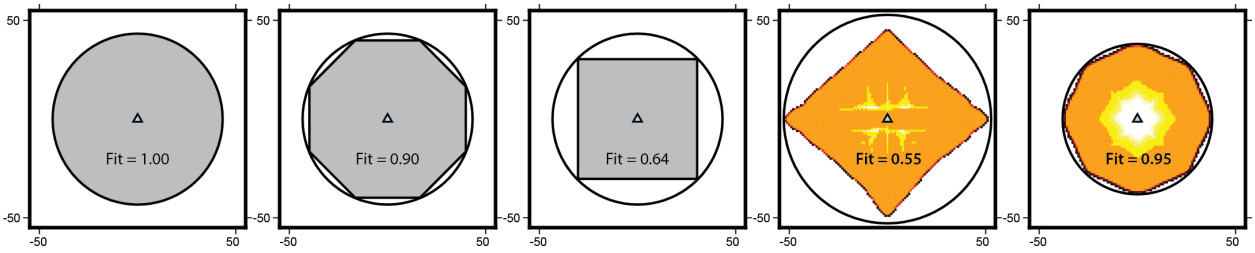

Figure 6: Fitness scores of different shapes. From left to right: A circle has a perfect fit score of 1.00; an octagon has 0.90 times the area of a circumscribing circle; a square has a score of 0.64; Two flat surface tests for slope proportional spreading algorithms with parent rules. The flow second to the right is 4-connected (4/P/S) and has a score of 0.55, while the rightmost flow is 8-connected (8/P/S) with a score of 0.95. While the 4/P/S flow scores worse than a square, its 8-connected version passes the test as it scores better than an octagon.

3.3.1 Benchmark Parameters

The DEM used in this benchmark is a horizontal plane (all grid locations have the same elevation) with a spatial resolution of 1 m. A single vent is located at the DEM center with a total volume of 1000 m^3, and a pulse volume of 1 m^3. When the simulation is finished, model output is used to find the inundated cell farthest from the vent (d_{max}). The total inundated area is divided by the area of a circle with radius d_{max} to provide the Fit score.

3.3.2 Results

Five of the eight algorithms unambiguously passed the test of performing better than an octagon. In this test 8-connected algorithms outperformed 4-connected algorithms. Four algorithms unambiguously passed this test: 8/P/S, 8/N/S, 4/N/E, and 8/N/E.

Bingham Circle Results

Algorithm	Circularity	
4/P/S	0.55	Worse than a square.
8/P/S	0.95	Better than an octagon.
4/N/S	0.55	Worse than a square.
8/N/S	0.98	Better than an octagon.
4/P/E	0.77	Between a square and an octagon.
8/P/E	0.80	Between a square and an octagon.
4/N/E	1.00	Perfectly circular.
8/N/E	0.99	Better than an octagon.

3.4 Level 3: Replication of real lava flows over complex topography

The recent availability of global or near-global topographic datasets, such as SRTM or ASTER GDEM has enabled the direct observation of the underlying surface of even more recent lava flows. Flow algorithms can be validated against recent lava flows by simulating lava over these surfaces with parameters defined by the new lava flows. The 2012-3 Tolbachik lava flows will be used as a benchmark for the eight flow algorithms. As discussed above (Section 1.1), the earliest lavas flowed from a fissure to the west. Before later stage flows began moving to the east, the volume of the lavas were 0.22 km^3. The modal thickness of the flow has been found to be 7.8 m and the areal extent was mapped with orthoimages [Kubanek et al., 2015].

For this example, two metrics which are commonly employed to validate lava flow simulators against real flows will be used: model sensitivity and a fitness metric called the "Jaccard coefficient." An alternative bayesian approach to these metrics is discussed in Section 4. Model sensitivity is defined as

$$\text{Model Sensitivity} = \frac{|Lava \cap Sim|}{|Lava|} \tag{7}$$

where $|Lava \cap Sim|$ is the size of the interesection of the simulation and the true lava flow (the True Positives) and $|Lava|$ is the size of the lava flow. This gives a percentage of the true lava flow that the simulation correctly predicted.

The Jaccard coefficient, or fit, is defined as

$$\text{Jaccard Fit} = \frac{|Lava \cap Sim|}{|Lava \cup Sim|} \tag{8}$$

where $|Lava \cup Sim|$ is the size of the union of the lava flow and a simulated flow. This gives a percentage of the total area covered by either the simulated flow or the true flow that is covered by both.

Each flow algorithm is run using the following parameters. Flows are run over both 3-arcsecond SRTM topography (75 m grid resolution at 56°N) and bistatic TanDEM-X topography processed by Kubanek et al. [2015]. The pulse volume, the volume added to vent cells at each code loop (i.e. each instance of the PULSE module), is set at the product

of the grid-cell area (5600 m^2 for the SRTM DEM and 225 m^2 for the TanDEM-X DEM) and the residual thickness (7.8 m). Both fitness metrics are calculated with the resulting model output, given as a list of inundated locations. "Failure" can be defined here as either metric falling below 50% for the sake of example.

Tolbachik Validation Flow Parameters

Elevation Model	75-m SRTM or 15-m TanDEM-X
Residual Thickness	7.8 m
Pulse Volumes	44200 m^3 (SRTM) or 1800 m^3 (TanDEM-X)
$Vent_N$ Easting	582800 m (UTM Zone 57)
$Vent_N$ Northing	6182100 m
$Vent_N$ Total Volume	$4.63 \cdot 10^7$ m^3
$Vent_S$ Easting	582475 m
$Vent_S$ Northing	6180700 m
$Vent_S$ Total Volume	$1.737 \cdot 10^8$ m^3

3.4.1 Results

Three example algorithms are illustrated in Figure 7. One primary observation is that all simulations had a longer run-out length on the finer TanDEM-X grid than on the coarser SRTM grid. Despite this, all flows took the correct major pathways taken by the true lava flow. A small diagram in the top left corner of each map in Figure 7 shows the true positives, false positives, and false negatives in each simulation. True positives are areas inundated by both flow and simulation, false positives are areas simulated as being inundated but are not mapped as such, and false negatives are areas hit by lava that the simulation failed to forecast.

The best algorithm and DEM pair were the 8/N/S algorithm over SRTM (top left of Figure 7), while this same algorithm performed fairly poorly over TanDEM-X topography. For the SRTM simulation, this algorithm achieved a model sensitivity of 82.8% and a Jaccard fitness score of 63.1%. Graphically, sensitivity is calculated as the green area in the Figure 7 diagram divided by the green and red areas. The Jaccard fitness is the green area divided by the total area of the diagram. Because the Jaccard fitness statistic essentially expands the denomenator of model sensitivity, it will never be higher than model sensitivity.

If success and failure are defined by having a fits of greater or less than 50%, all models tested would pass for the SRTM DEM and about half would pass for the TanDEM-X DEM. All but one model (4/P/E) performed worse on the TanDEM-X DEM. The Jaccard fit and Sensitivity for all models are given below.

Tolbachik Flow Results

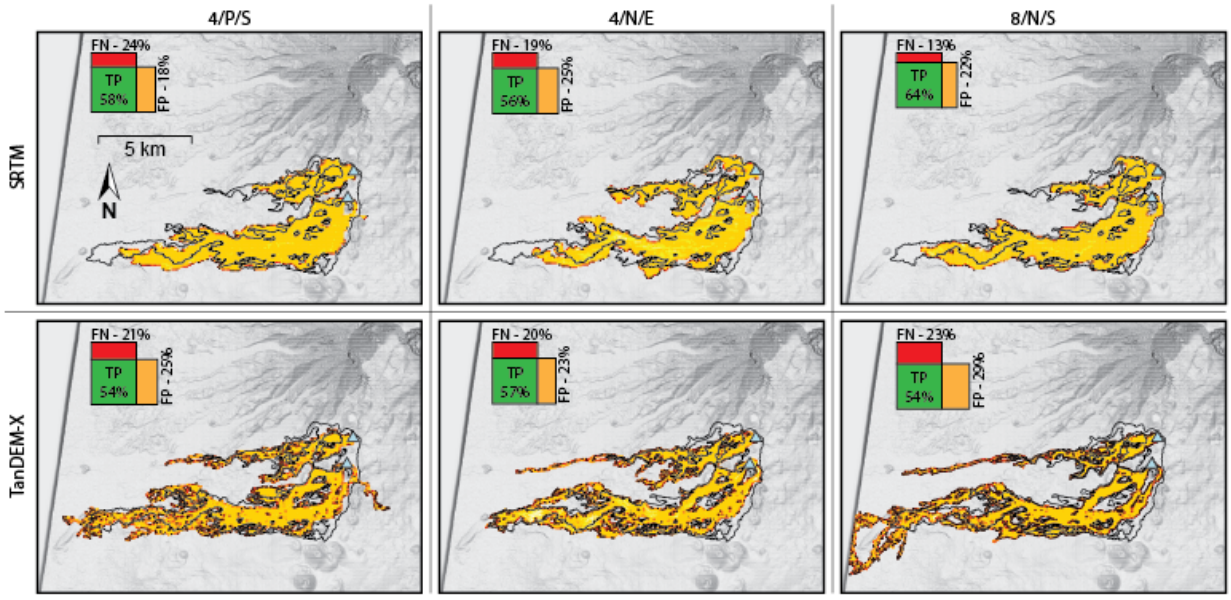

Figure 7: Three simulation algorithms (4/P/S, 4/N/E, and 8/N/S) applied to two elevation models (SRTM and TanDEM-X) to simulate the 2012-3 Tolbachik lava flows, outlined in black. Lava is emitted from two vent locations marked as blue triangles in the simulation. Diagrams showing the relative True Positives (green), False Positives (orange), and False Negatives (red) are illustrated in the top left of each simulation.

Transition	**SRTM DEM**		**TanDEM-X DEM**	
Function	Jaccard	Sensitivity	Jaccard	Sensitivity
4/P/S	56.7%	76.4%	53.0%	72.4%
8/P/S	61.1	80.8	46.8	67.2
4/N/S	57.2	77.5	44.0	64.0
8/N/S	63.1	82.8	46.7	67.4
4/P/E	51.2	71.5	54.2	73.4
8/P/E	58.8	78.2	56.3	76.0
4/N/E	54.5	74.5	55.7	73.7
8/N/E	59.6	78.8	56.2	75.3

4 Bayesian Applications for Lava Flow Models

The final step Bayarri et al. [2007] give for validating computer models is "feeding [observations and results] back to revise the model."

The use of computer models to forecast hazards is a fundamentally Bayesian strategy: there is an initial concern due to hazards and computer models help us inform, constrain, and update this concern. Using Bayesian statistics can therefore be an improvement in testing lava flow models, over the two commonly used fitness tests, model sensitivity and the Jaccard index, because of their more direct application to informing perceived risk.

Three tools will be used in this section: A posterior probability, a "negative" posterior probability, and a Bayes factor. Bayes theorem connects a phenomenon A to observations B through the function

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \tag{9}$$

where $\Pr(A)$ is the general probability of A occuring, $\Pr(B)$ is the probability of B being observed, and $\Pr(B|A)$ is the conditional probability of B given the occurence of A. $\Pr(B|A)$ is also known as model sensitivity, which is a common fitness statistic and was discussed in Section 3.4

The left side of Equation 9, $\Pr(A|B)$, is the Posterior probability of A and can be stated as "the probability that lava will inundate a location if the model forecasted inundation at that location." A second posterior, which I call the negative posterior, is $\Pr(\neg A|\neg B)$ and is the obverse of $\Pr(A|B)$. This negative posterior relates not being inundated by lava at a given location to a safe outcome forecasted by a simulation and can be calculated by modifying Equation 9 and substituting A for $Lava$ (the lava flow) and B for Sim (the simulation), resulting in the formula

$$\Pr(\neg Lava|\neg Sim) = \frac{\Pr(\neg Sim|\neg Lava)\Pr(\neg Lava)}{\Pr(\neg Sim)} \tag{10}$$

where the $\neg$ symbol indicates the event or observation not happening, and $\Pr(\neg Sim|\neg Lava)$ is model specificity.

The negative posterior is important in hazard forecasting as it is in some sense a probability of safety. The more common posterior $\Pr(A|B)$ (or $\Pr(Lava|Sim)$) does not contain information about areas that the simulation does not inundate; while it can support the hypothesis that lava will hit a location given a simulated hit, it cannot estimate one's relative risk if the simulation forecasts a safe outcome. The negative posterior $\Pr(\neg Lava|\neg Sim)$ does just this, and informs a user whether to rely on a safe outcome from a simulation. If, for example, the posterior probability $\Pr(Lava|Sim)$ is high while the negative posterior probability $\Pr(\neg Lava|\neg Sim)$ is low for a given lava flow simulator, areas that are evacuated due to the simulation outcome will be evacuated for good reason, but many areas will likely be inundated that were not evacuated due to the simulation outcome. This is why it is important to estimate and ultimately try to improve both posterior metrics.

Bayes factors provide a tool to test the relative likelihood of a hypothesis against another. Aspinall et al. [2003] introduced this tool to volcano hazard forecasting by testing whether the onset of particular seismic events before the 1993 Galeras catastrophe was a significant indicator of the eruption or not. A Bayes Factor (BF) relating two models is given by Jeffreys [1998] as

$$\text{BF} = \frac{\Pr(\text{Data}|\text{Model 1})}{\Pr(\text{Data}|\text{Model 2})} \tag{11}$$

In the example of Galeras, the "data" are the seismic events, Model 1 is "imminent explosion," and Model 2 is "not imminent explosion" [Aspinall et al., 2003]. Below, I will apply this with the data being the probability of simulated inundation and the models "lava inundation" and "not lava inundation." Jeffreys [1998] provided a log-scale interpretation to the value of BF in Equation 11, given in the Table 2.

Table 2: Bayes Factor Interpretations (modified from Aspinall et al. [2003])

BF Value	Description
$BF > 10^2$	Evidence for Model 1 is Decisive.
$10^{1.5} < BF < 10^2$	Evidence for Model 1 is Very Strong.
$10^1 < BF < 10^{1.5}$	Evidence for Model 1 is Strong.
$10^{0.5} < BF < 10^1$	Evidence for Model 1 is Substantial.
$10^0 < BF < 10^{0.5}$	Evidence for Model 1 is just worth a mention.
$10^{-0.5} < BF < 10^0$	Evidence for Model 2 is just worth a mention.
$10^{-1} < BF < 10^{-0.5}$	Evidence for Model 2 is Substantial.
$10^{-1.5} < BF < 10^{-1}$	Evidence for Model 2 is Strong.
$10^{-2} < BF < 10^{-1.5}$	Evidence for Model 2 is Very Strong.
$BF < 10^{-2}$	Evidence for Model 2 is Decisive.

In the same manner as the final validation level, the statistics discussed above will be calculated based on the areal extent of flows and simulations. The probability of the lava flow inundating an area N can be given as

$$\Pr(A) = \frac{|Lava|}{|N|} \tag{12}$$

where $|Lava|$ is the areal size of the flow (i.e. literally the number of DEM grid cells the lava inundates) and $|N|$ is the size of the area of interest, or the potential hazard area. The probability of the simulation is similarly found to be

$$\Pr(Sim) = \frac{|Sim|}{|N|}. \tag{13}$$

By substituting these definitions and model sensitivity (Equation 7, $|Lava \cap Sim|/|Lava|$) in Equation 9, the posterior probability of lava flow inundation, given a simulation that forecasts inundation can be recast as

$$\Pr(Lava|Sim) = \frac{\frac{|Lava \cap Sim|}{|Lava|}\frac{|Lava|}{|N|}}{\frac{|Sim|}{|N|}} \text{ , or simplified,} \tag{14}$$

$$= \frac{|Lava \cap Sim|}{|Sim|}. \tag{15}$$

where $|Lava \cap Sim|$ is the size of the intersection of the lava flow and simulated flow (again, the number of DEM grid cells). Note that this posterior probability is independent of the potential hazard area.

The negative posterior can be stated in terms of the sizes of the lava flow and simulated flow as well.

$$\Pr(\neg Lava|\neg Sim) = \frac{|\neg Lava \cap \neg Sim|}{|\neg Sim|} \tag{16}$$

Calculating the size or number of grid cells of $\neg Lava$ or $\neg Sim$ is fundamentally dependent on the potential hazard area, as $|\neg Lava|$ is defined as

$$|\neg Lava| = |N| - |Lava|. \tag{17}$$

Because of this, we must define the size of the potential hazard area N ($|N|$).

Potential Hazard Area There are multiple strategies to estimating an *a priori* hazard area. Kauahikaua et al. [1995] for instance identified catchments or "lava sheds" in which a volcanic vent was erupting, and identified these lava sheds as the hazard area. Kilburn [2000] provided a theoretical maximum distance that a lava flow can travel given the mass flux of magma erupting at the vent location. A combination of these two would provide an objective hazard area defined as the area within the "Kilburn distance" that is topographically below the volcanic vent. The theoretical maximum distance, or hazard radius, given by Kilburn [2000] is

$$R_{max} = \sqrt{\frac{3\epsilon SQ}{\rho g \kappa}} \tag{18}$$

where ϵ is an empirical value related to the amount of extension of lava crust allowed before it fails (10^{-3}), S is the tensile strength of this crust (10^7 Pa), ρ is the lava crust density (2200 kg m^{-3}), g is gravitational acceleration, κ is the bulk thermal diffusivity (4×10^{-7} m^2 s^{-1}) and Q is the mean volumetric flow rate from the vent. From this, the hazard radius for the Tolbachik 2012-3 flow is calculated to be 39 km, given a magma flux of 440 m^3 s from the vent as was estimated early in the eruption [Belousov et al., 2015]. The total area within this radius that is also below the vent-plus-modal-flow-thickness elevation is 1,415 km^2. Note that the mapped flow area of 26 km^2 only covers 1.9% of this defined hazard area (i.e. $\Pr(Lava) = 0.019$).

A second strategy would be to run many lava flow models from the known vent location(s) while varying input parameters. This would give a range of flows and the true flow might be completely contained within the region given by this range of simulations. Below, a Monte Carlo (MC) method will be used to simulate a large range of flows. If we define a potential hazard area as any location inundated by at least one simulated flow in this MC approach, the hazard area would be 72 km^2. As the mapped flow area from the Tolbachik eruption is 39% of this area, it would be more practical to use this as the *a priori* hazard area because it more reasonably reflects the potential inundation area. Both the Kilburn-Kauahikaua method and this MC method are illustrated in Figure 8.

Review of Validation Level 3 Instead of using model sensitivity and the Jaccard index as benchmarks for the various lava flow models, now the two posteriors will be used. The potential hazard area is defined as the distribution of MC simulations (72 km^2). To give an example calculation, $\Pr(Lava|Sim)$ is found with Equation 15 by dividing true positives (green boxes in 7) by the simulation area (green and red boxes). The negative posterior, $\Pr(\neg Lava|\neg Sim)$, is found by diving true negatives (blue area, in bottom right diagram of Figure 8), by the area not simulated (top half of bottom right diagram of Figure 8).

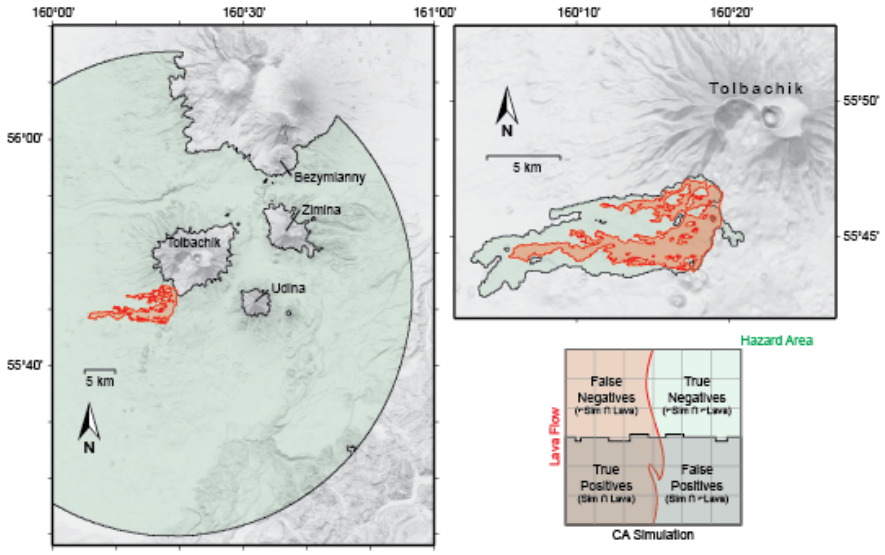

Figure 8: Potential hazard areas of the 2012-3 Tolbachik Flows in light green, defined by a maximum flow radius (left, Equation 18) and the total areal coverage of 100,000 random flow simulations (top-right, Section 4.2). The mapped lava flow is red. The chart to the bottom right shows four regions defined by the presence or absense of lava and simulated lava across a map grid.

Traditional Fit Metrics and Bayesian Posterior Functions

Transition Function	Results of simulations over SRTM DEM			
	Jaccard	Sensitivity	$\Pr(Lava\|Sim)$	$\Pr(\neg Lava\|\neg Sim)$
4/P/S	56.7%	76.4%	70.4%	84.5%
8/P/S	61.1	80.8	73.2	87.3
4/N/S	57.2	77.5	70.3	85.1
8/N/S	63.1	82.8	74.4	88.5
4/P/E	51.2	71.5	66.0	81.4
8/P/E	58.8	78.2	72.0	85.7
4/N/E	54.5	74.5	68.7	83.3
8/N/E	59.6	78.8	72.8	86.2

For the rest of this section, the 8/N/S (the 8-connected, no parent-child relationships, with slope-proportional spreading) algorithm will be used. This is preferred because it outperformed other algorithms over the SRTM DEM. By applying this model to the Tolbachik lava flows, Bayesian methods will be used to improve the "Pulse Volume" parameter and will later be used to constrain model uncertainty at Tolbachik.

4.1 Improving model performance on one model parameter

In the Tolbachik benchmark tests given as examples of comparing simulation algorithms against real lava flows (Section 3.4), all but one algorithm performed worse on the TanDEM-X derived elevation model. This was in part due to large run-out distances in the simulations (e.g. bottom right of Figure 7), which considerably increased simulation false positives. The

large run-out distances might be due to the pulse volume, the volume of lava given to source cells at each code loop in MOLASSES, being poorly chosen. Here, the Bayesian statistics defined above will be used to compare different pulse volumes and identify an optimal pulse volume.

An optimal pulse volume will ideally produce a flow simulation with the highest posterior and negative posterior value. Pulse volumes with high associated posterior values will produce simulations where areas simulated as inundated by lava will have a high likelihood of actually being inundated by lava. Pulse volumes with high associated negative posterior values will produce simulations where areas simulated to not be inundated will hava a high likelihood of actually not being inundated.

4.1.1 Model Execution

To populate *Sim*, I have run the MOLASSES lava flow code using TanDEM-X derived parameters listed in Table 3. All variables are fixed except the pulse volume parameter, which is the amount of lava delivered to source cells in the Cellular Automata grid of MOLASSES. The lowest pulse volume, 1755 m^3 per pulse, is approximately the product of the TanDEM-X DEM grid cell size (225 m^2) and the residual flow thickness (7.8 m). The other 15 pulse volumes are multiples of this volume (i.e. they are 1.5 to 8.5× 1775 m^3).

Table 3: MOLASSES Flow Parameters

Elevation Model	15-m bistatic TanDEM-X, 11 Nov 2015
Modal Thickness	7.8 m
Pulse Volumes	16 equally separated volumes, [1755,14917] m^3
$Vent_N$ Easting	582800 m (UTM Zone 57)
$Vent_N$ Northing	6182100 m
$Vent_N$ Total Volume	$4.63 \cdot 10^7$ m^3
$Vent_S$ Easting	582475 m
$Vent_S$ Northing	6180700 m
$Vent_S$ Total Volume	$1.737 \cdot 10^8$ m^3

Model output is compared to a list of x,y locations in the Tolbachik area that have been inundated or not. This location list is stored in a raster with the same projection and extent as the elevation model used in MOLASSES. ASCII locations output by MOLASSES are also listed in the same projection within the same extent as the elevation model. This enables direct comparison between the Model information (i.e. *Sim*) and the mapped lava flow (i.e. *Lava*). True Positives, False Positives, and False Negatives are reported as cell counts (number of grid locations where *Lava* and *Sim* agree or not). Three examples of these simulations are mapped in Figure 9.

4.1.2 Results

Three example simulations are shown in Figure 9 using simulation parameters from Table 3 and different pulse volume values. With increased pulse volume, simulated run-out distance

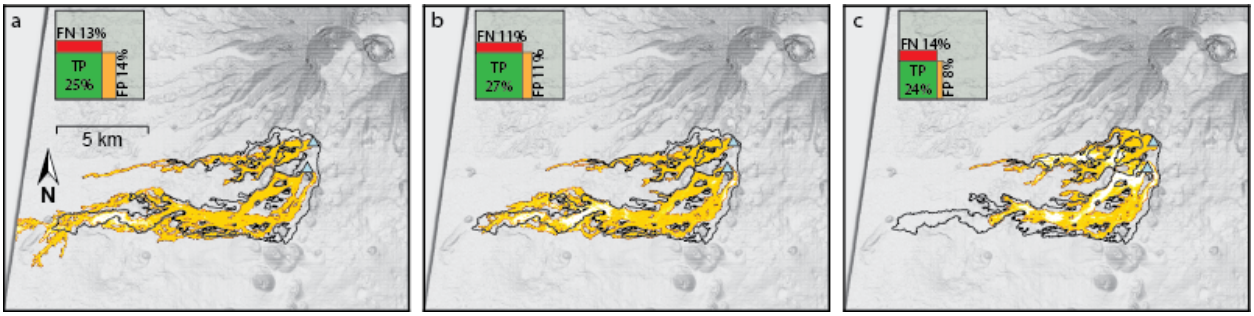

Figure 9: MOLASSES Simulations of the 2012-3 Tolbachik Lava Flows. Vents are shown as blue triangles and the mapped flow is outlined in black. a) Pulse Volume = 1755 m^3, the simulation far exceeds the true runout distance; b) Pulse Volume = 4387 m^3, this simulation performs best under the negative posterior $\Pr(\neg Lava|\neg Sim)$ test; c) Pulse Volume = 14040 m^3, this simulation performs best under the posterior $\Pr(Lava|Sim)$ test, but does not have a runout length similar to the mapped flow.

is shorter. This is because the MOLASSES code ends once all volume is delivered to the vents and the DISTRIBUTE module has run once more. In other words, if the pulse volume is doubled, the number of times the PULSE and DISTRIBUTE modules will be run will be halved, as the total volume will be delivered to the vents in half the code loops (see Figure 1). By running DISTRIBUTE fewer times, cells have fewer opportunities to advect lava downslope.

The posterior statistical measure is the fundamental tool of Bayesian statistics, and quantifying it enables an update of belief in risk of lava inundation. A perfect posterior value would mean that if the model simulates lava inundating a location, lava will certainly inundate that location. The posterior is calculated for simulated lava flows of different Pulse Volumes and is graphed in Figure 10. From this, it can be seen that the highest pulse volumes, which coincidentally form the shortest flow simulations, perform best with this test, with the best fit having a pulse volume of 14040 m^3 per algorithm loop (Figure 9,c). A local maximum does exist in the low pulse volumes at 4387 m^3 per loop.

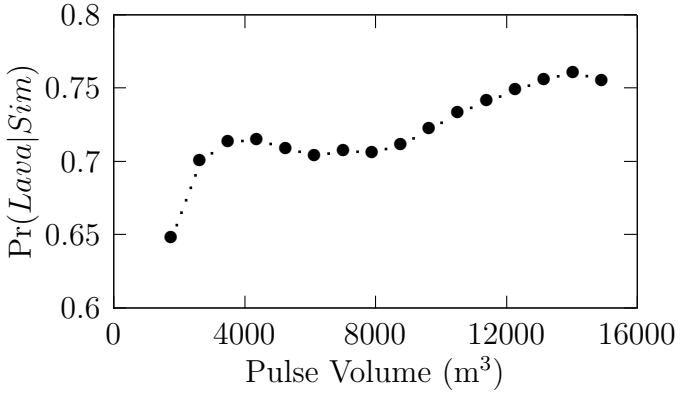

Figure 10: Posterior $\Pr(Lava|Sim)$ for MOLASSES flows with differing Pulse Volumes.

The negative posterior $\Pr(\neg Lava|\neg Sim)$, is the percentage of non-inundated area in the simulation that is also not inundated in real life. A perfect negative posterior would indicate that, if a model does not simulate a hit for a location, lava will certainly not inundate that location. The negative posteriors of simulations with different pulse values are shown in Figure 11. Unlike the previous posterior analyzed, the best performing flows have smaller pulse volumes and the best performing volume is 4387 m^3 per model pulse loop (Figure 9,b).

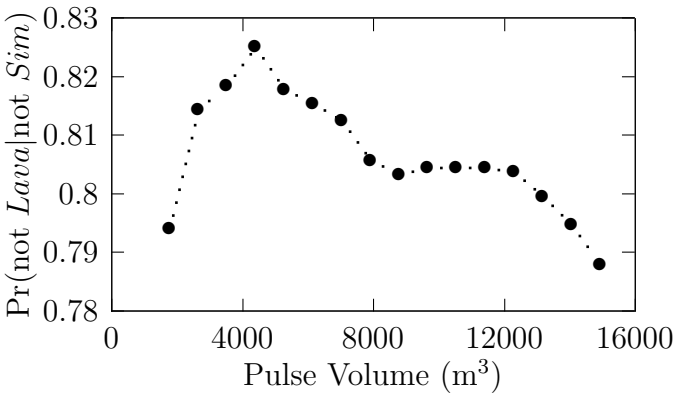

Figure 11: Negative posterior $\Pr(\neg Lava|\neg Sim)$ for MOLASSES flows with differing Pulse Volumes.

4.2 Incorporating Model Uncertainty with a Monte Carlo method

Model uncertainty is a result of input parameter uncertainty, such as uncertainty in the underlying DEM. This can be distinguished from model error, which might be defined as the difference between the true lava flow and a simulation carried out with perfect input parameters, and is created by the inherent deviations between a computer model and real life processes. Because there is parameter uncertainty, it is essential to quantify the range of model solutions given the likely range of each parameter.

In this example, elevation uncertainty will be examined. Elevation uncertainty is an element in the MOLASSES module **INITFLOW**, where each grid cell elevation can be defined randomly before the lava flow simulation begins. The user can add an elevation uncertainty, in meters, to the configuration file. If this value is provided, each grid cell will receive a new elevation value randomly selected from a normal distribution whose mean is the DEM elevation and the standard deviation is the uncertainty value.

The Monte Carlo method runs MOLASSES 100,000 times over a 75-m SRTM DEM. Vertical uncertainty of this data is estimated by Rodríguez et al. [2006] for Eurasia to be 6.2 m at a 90% confidence level and is shown to be randomly distributed. With this result, elevation uncertainty in the MOLASSES model is given a value of $1\sigma = 3$ m. Other input parameters remain unchanged from the benchmark exercise above; MOLASSES flow parameters for the Monte Carlo model are listed in Table 4. The combined 100,000 simulations are mapped in Figure 12 where flow color indicates the number of flows that impacted each location.

Table 4: Monte Carlo MOLASSES Flow Parameters

Elevation Model	75-m SRTM
Elevation Uncertainty, 1σ	3 m
Residual Thickness	7.8 m
Pulse Volume	44200 m^3
$Vent_N$ Easting	582800 m (UTM Zone 57)
$Vent_N$ Northing	6182100 m
$Vent_N$ Total Volume	$4.63{\cdot}10^7$ m^3
$Vent_S$ Easting	582475 m
$Vent_S$ Northing	6180700 m
$Vent_S$ Total Volume	$1.737{\cdot}10^8$ m^3

4.2.1 Bayesian distribution of MC results

The reliability of a model can be better understood by showing the distribution of model performance given model uncertainty, as opposed to treating model parameters, and thus model output, as completely certain. Figure 13 shows the distribution of the posterior and the negative posterior scores. Each dot in the main chart of Figure 13 represents a single flow simulation over a partially randomly generated DEM. The clustering of these points shows a positive correlation between the two posterior metrics, and both metrics are not normally distributed as shown on the histograms on either side of the main chart of Figure 13. The flow simulation assuming no elevation uncertainty (shown in the top right corner of Figure 7) fits the mapped flow better than the median value of both posteriors in the distribution, though it still lays within the MC distribution.

If the elevation model used were perfect, the posterior values would be a single number ($\Pr(Lava|Sim) = 74.4\%$ and $\Pr(\neg Lava|\neg Sim) = 88.5\%$). However, because elevation values have inherent uncertainty, model fitness, as defined by the posterior values, can be given as a range. Including elevation uncertainty, the $\Pr(Lava|Sim)$ fitness has a range of 59-76% and the $\Pr(\neg Lava|\neg Sim)$ has a range of 77-91%.

4.2.2 Estimating inundation risk from the simulated frequency of inundation

Figure 12 shows a map view of the probability of inundation from the 100,000 MC simulations. Generally, areas within the mapped flow appear to be inundated by more simulations than outside the mapped flow. But can the probability of simulation inundation, $\Pr(Sim)$, be used in a more formal way to judge the probability of lava inundation? In this section, the Tolbachik region map will be split into sub-regions based on $\Pr(Sim)$, and the probabilities of inundation and not inundation will be compared.

Three example sub-regions are shown in Figure 14. These sub-regions are defined as the area where $\Pr(Sim)$ falls between a 10% range. For instance, the top sub-region in Figure 14 shows all locations that were forecast as inundated by at least 90% of all MC simulations, while the middle sub-region example contains all locations inundated by 40-50% of simulations. The sub-regions indundated by 0-10% and 90-100% of simulations are

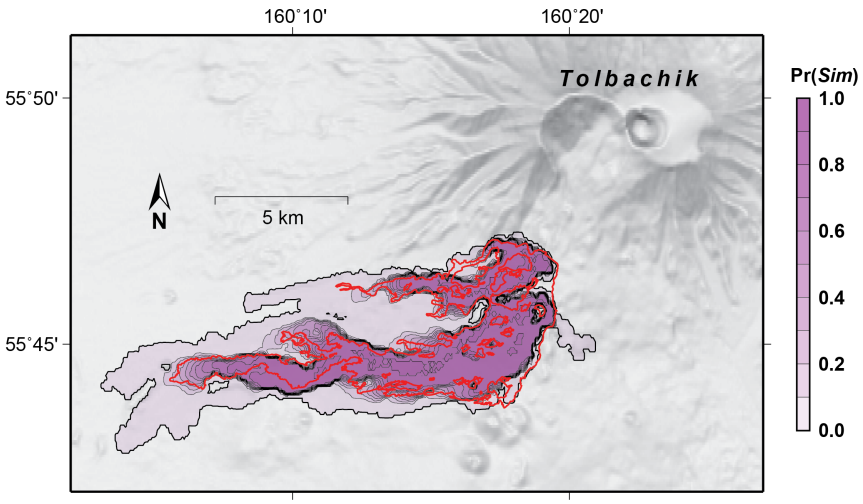

Figure 12: Cumulative distribution of 100,000 simulated lava flows over SRTM topography with 3 m elevation uncertainty. The red outline is the mapped flow extent of the 2012-3 Tolbachik flow. Darker purple areas represent more simulation hits (i.e. higher Pr(Sim)).

the largest sub-regions, while other sub-regions are thin rings between these two bounding sub-regions.

It can be seen in Figure 14 that most of the mapped flow is covered by the Pr($Sim \geq 90\%$) sub-region, as the green true positive region is larger than the red false negative region. The sub-region defined as Pr($0 < Sim < 10\%$) does not spatially intersect with the mapped flow area as much as the Pr($Sim \geq 90\%$) sub-region, and this can be seen in the lower right of Figure 14 as the sub-region area is mostly orange false positives with small green true positive areas.

The relative risk of inundation can be calculated for each subregion by comparing the probability of actual flow inundation Pr($Lava$) against the probability of not inundation Pr($\neg Lava$) using the Bayes Factor of Equation 11. Here "Data" is the probability of simulated inundation Pr($Sim = \mathrm{X}$), Model 1 is Real Flow Inundation and the opposing Model 2 is No Real Flow Inundation. For example, the relative probability of inundation for the sub-region defined by Pr($40 \leq Sim < 50\%$) can be given as

$$\mathrm{BF} = \frac{\Pr(0.4 \leq Sim < 0.5 \mid \text{Inundation})}{\Pr(0.4 \leq Sim < 0.5 \mid \text{Not Inundation})}. \tag{19}$$

The numerator Pr($0.4 \leq Sim < 0.5 \mid$ Inundation) is the probability of 40-50% of simulations hitting a given location, given the location actually being hit by lava. Graphically, this is the percent of the true flow (red and green areas in the center example of Figure 14) that are within the simulated sub-region (green areas in the center example of Figure 14). The denominator Pr($0.4 \leq Sim < 0.5 \mid$ Not Inundation) is probability of 40-50% of simulations hitting a location, given the location is not inundated in the real flow. This is the percent of the area not hit by the flow (light gray and orange areas in Figure 14) that is within the subregion (orange areas in Figure 14).

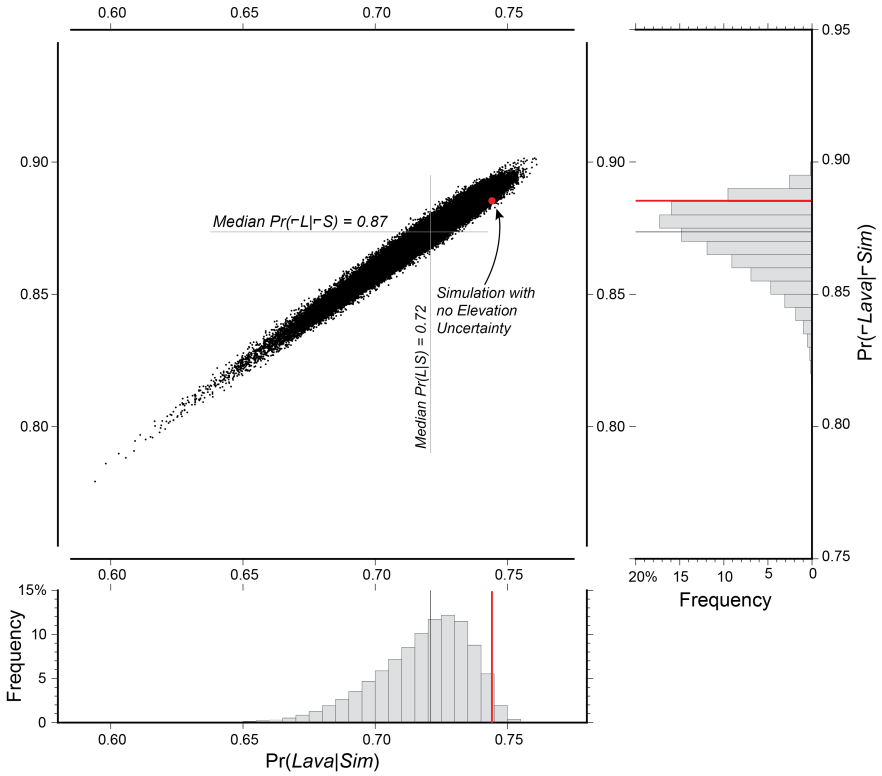

Figure 13: Fitness statistic distribution between for 100,000 simulations of the 2012-3 Tolbachik Lava Flows, over SRTM topography with 3 m standard elevation uncertainty. Each point represents the posterior probabilities of inundation/non-inundation for one simulation. Red lines are the fitness values of a simulated flow over SRTM data assuming 0 m elevation uncertainty (top right corner of Figure 7). Black lines are placed at the median values of each posterior probability.

The probability that 40-50% of simulations hit a location that is inundated is 2.7%. The probability the 40-50% of simulations hit a location that is not inundated by lava is 2.1%. The Bayes Factor is then calculated to be 1.3, where the model of Inundation is 1.3 times more likely to describe this subregion than the model of Not Inundation. Referring to Table 2, this result means that the preference for Inundation over Not Inundation is "just worth a mention." Results for each 10% wide sub-region are given in Table 5 and are illustrated in Figure 15.

Only the sub-region least likely to be hit by simulations, where $\Pr(Sim < 10\%)$, has a Bayes Factor of $< 10^{-1}$, which is interpreted as strong evidence against lava flow inundation. As the Bayes Factor can be treated as posterior odds for or against a model [Aspinall et al., 2003], a factor of 1/10 indicates 10:1 odds against inundation. Sub-regions with factors greater than 1/10, and are therefore more in support of flow inundation, have odds less than 10:1 against inundation. As 10:1 odds against inundation is the same as a probability of 9% for inundation, all sub-regions besides the $\Pr(Sim < 10\%)$ sub-region have $\Pr(Lava > 9\%)$.

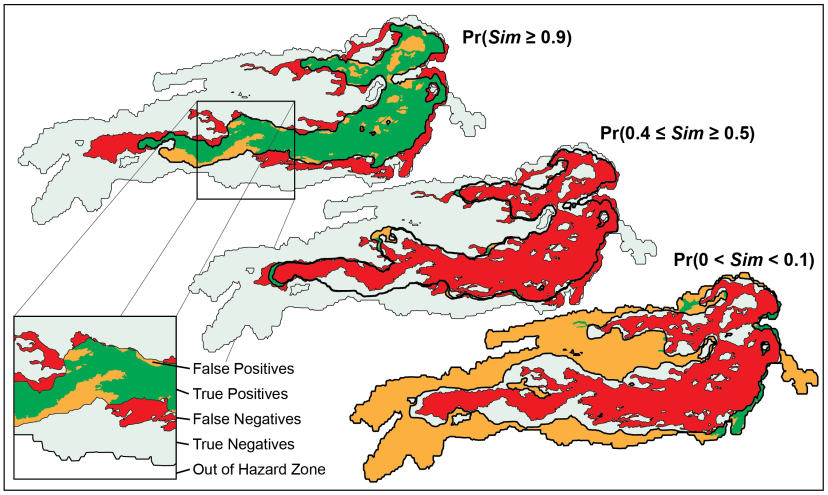

Figure 14: Sub-regions within the Monte Carlo solution, defined by Pr(Sim). At the top, all areas inundated by ≥90% of all simulations are shown in green (True Positives) and orange (False Positives). The remaining mapped lava flow (False Negatives) is red. The underlying region is the total Monte Carlo Hazard Area. At center, A thin band represents all areas inundated by 40-50% of all Simulations. At bottom, a thick shell of areas rarely inundated by simulated lava is mostly orange, indicating rarely simulated flow areas are unlikely to have been inundated in real life.

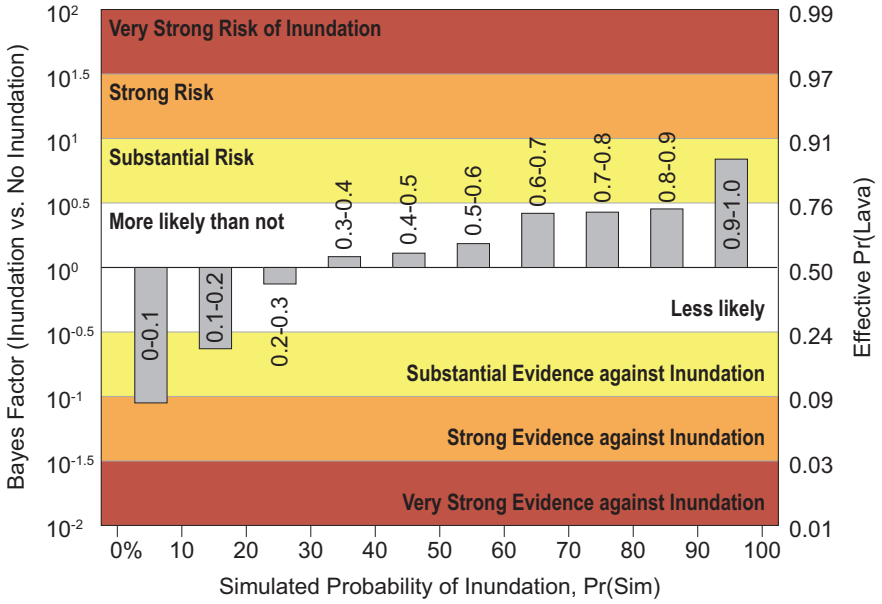

Figure 15: Relative likelihood of inundation given Pr(Sim). Only locations inundated by $< 10\%$ of Monte Carlo simulations show strong evidence against inundation in the actual 2012-3 Tolbachik lava flows.

5 Discussion

5.1 Benchmark Validation

Unique flow algorithms are validated by using common benchmarks that show whether these algorithms replicate expected lava flow morphologies under specific conditions. Different

Table 5: Relative likelihood of inundation given Pr(Sim)

X	Pr(Sim=X| $Lava$)	Pr(Sim=X| $\neg Lava$)	Bayes Factor	Jeffreys [1998] Interpretation
0 <X< 0.1	0.06	0.67	0.09	Strong evidence against inundation
0.1 ≤X< 0.2	0.02	0.08	0.23	Substantial evidence against inund.
0.2 ≤X< 0.3	0.03	0.04	0.74	No Inundation more likely
0.3 ≤X< 0.4	0.03	0.03	1.21	Inundation more likely than not
0.4 ≤X< 0.5	0.03	0.02	1.29	Inundation more likely than not
0.5 ≤X< 0.6	0.03	0.02	1.53	Inundation more likely than not
0.6 ≤X< 0.7	0.04	0.01	2.63	Inundation more likely than not
0.7 ≤X< 0.8	0.05	0.02	2.69	Inundation more likely than not
0.8 ≤X< 0.9	0.06	0.02	2.84	Inundation more likely than not
0.9 ≤X≤ 1.0	0.66	0.10	6.93	Substantial evidence for inundation

benchmark tests that are applicable to CA codes fall into three validation levels: 1) tests that show that simulations don't change when parameters remain meaningfully identical; 2) tests that show that simulations replicate experimental results or analytical expectations; and 3) tests that show that simulations replicate real world flow examples. A summary of the results of the eight algorithms against the example benchmarks are given below in Table 6.

Table 6: Transition Algorithm Results

Transition Function	**Levels** 1 DEM Rotation	2 Pancake	3 SRTM	3 TanDEM-X
4/P/S	Pass	Fail	Pass	Pass
8/P/S	Pass	Pass	Pass	Fail
4/N/S	Fail	Fail	Pass	Fail
8/N/S	Pass	Pass	Pass	Fail
4/P/E	Fail	Ambiguous	Pass	Pass
8/P/E	Fail	Ambiguous	Pass	Pass
4/N/E	Fail	Pass	Pass	Pass
8/N/E	Fail	Pass	Pass	Pass

Because these validation levels increase in complexity from Level 1 to Level 3, one possible strategy in validating different algorithms would be to only test algorithms at more complex benchmarks after they successfully pass less complex benchmarks. Valid models might then be determined by elimination. Only 3 of 8 tested algorithms pass the first "rotating slope" benchmark: 4/P/S, 8/P/S, and 8/N/S. Although more algorithms passed the second "bingham flow on a flat surface" benchmark, only 8/P/S and 8/N/S passed the previous benchmark and this one. Both of these flows then successfully replicated the Tolbachik lava

flows over SRTM topography. Therefore the 8/P/S and 8/N/S algorithms hold up to three benchmark tests.

Overall, in all tests 8-connected models outperform 4-connected models. While equal sharing algorithms outperform slope-proportional sharing on a flat slope, they fail on a rotating DEM and perform about the same on real topography. There does not seem to be an unambiguously better choice between using parent-child relationships or not. If future tests continue to show similar performance between models with and without parentage, other reasons can be used to choose a model, such as computer run-time.

The strength of the MOLASSES code is that new algorithms, such as those used in the SCIARA model [Crisci et al., 2004], can be implemented relatively quickly and run through the Benchmarking tests, which are written in Python. Combinations of implementation strategies can also be created on the fly by adjusting the makefile of the MOLASSES code instead of the code itself.

5.2 Bayesian applications for real lava flows

Validation of lava flow models is important as a method of increasing the value of models to forecast lava flow processes, thereby decreasing preventable loss. Flow models are generally improved by reducing false positives and false negatives while increasing the true positive area, which is the union of a flow simulation and real, mapped lava flows. Often, reducing one type of error comes at the cost of increasing another type. For example, by increasing the pulse volume parameter, false positives can be reduced while false negatives are increased (Figure 9).

A decision can be made as to whether false positives or false negatives are more important to reduce in calibrating a lava flow model. One possible decision would be to prefer the reduction of false negatives over the reduction of false positives, with the reason being that a forecast of inundation without eventual engulfment by lava (a false positive result) is bad, but an unexpected engulfment by lava (a false negative) would be worse.

5.2.1 Using Bayesian statistics to compare models

In Section 4.1, an optimal pulse volume was defined as a volume that produced a simulated lava flow that had the highest posterior and negative posterior values. However, the pulse volume 14040 m^3 had the highest posterior value, while the pulse volume 4387 m^3 had the highest negative posterior value. The simple definition of "optimal" pulse volume does not seem to work.

The best pulse volume might be decided based on a preference for reducing false negatives as discussed above. The posterior probability of inundation, $\Pr(Lava|Sim)$ increases with increased true positives and decreases with increased false positives (Equation 15). It is essentially blind to false negatives. The negative posterior $\Pr(\neg Lava|\neg Sim)$ probability is the opposite, increasing with increased true negative results, while decreasing with increased false negative results (Equation 16).

If false negatives are more important to reduce than false positives, more weight can be given to the negative posterior value of a simulation than the positive posterior value. In Figure 16, the posterior and negative posterior values for each pulse volume (Figures 10

and 11) are used to produce weighted average scores of pulse volume. The extent to which false negative reduction might be preferred over false positive reduction is unknown, so four hypothetical weights are used. One of the weighted averages that is plotted assumes both posteriors are equally important in determining the best pulse volume parameter. The other three curves weight the negative posterior as being 2, 5, and 10 times as important as the positive posterior in scoring pulse volumes. Using these three weights, the highest scoring pulse volume is 4387 m^3 (Figure 9b). If the two posteriors have equal importance, the highest scoring pulse volume is 14040 m^3 (Figure 9c).

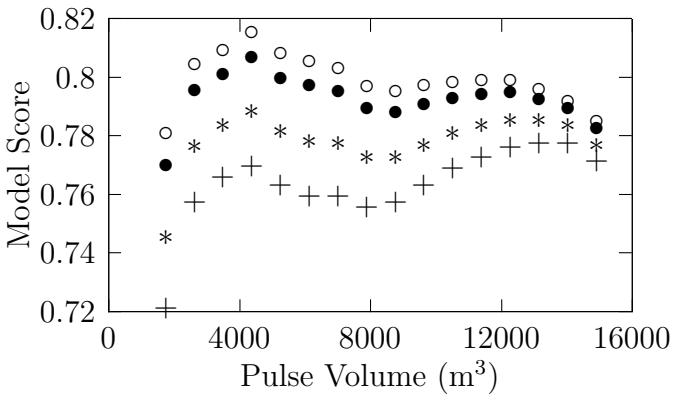

Figure 16: Weighted averages of Pr($Lava|Sim$) and Pr($\neg Lava|\neg Sim$) for different pulse volumes. Hollow circles, negative posterior (Pr($Lava|Sim$)) is given 10× the weight of the posterior (Pr($\neg Lava|\neg Sim$)); solid circles, 5× weight; asterisks, 2× weight; plusses, equal weight between posteriors. The highest scoring pulse volume is 4387 m^3 for all weight ratios except when both posterior values are weighted equally.

5.2.2 Decision Making with the Bayes Factor

By dividing the Monte Carlo set of simulations of the Tolbachik lava flows into sub-regions based on Pr(Sim), the relative likelihood of inundation was calculated (Figure 15). By using the Bayes Factor to compare likelihood of Indundation to Not Inundation, most sub-regions were not significantly better described by one model or the other. Only the sub-region of locations least likely to be hit by flow simulations had "strong evidence" against inundation, and only the sub-region of locations most likely to be hit by simulations had "substantial risk" of inundation. These results leave a large amount of ambiguity when making decisions to fortify or evacuate due to a lava flow.

In discussing the decision making process for calling an evacuation due to an eruption at Mount Vesuvius, Marzocchi and Woo [2007] provided a cost-loss model where two actions can be taken, protect and do not protect. Two costs are associated with these actions: $\mathcal{C}$ is the cost of protection and $\mathcal{L}$ is the cost of loss if the volcanic hazard occurs while a decision to not protect is made. If $\mathcal{L}$ is incurred, it is assumed to exceed cost $\mathcal{C}$, often because loss due to volcanic hazards includes the loss of life.

Marzocchi and Woo [2007] show that the minimum cost can be achieved if protection occurs when $p > \mathcal{C}/\mathcal{L}$ where p is the probability of the hazard occuring. The ratio $\mathcal{C}/\mathcal{L}$ is hard

to quantify because the socio-economic cost of lost lives or of lost trust in the government (in the case of false evacuations) is difficult to calculate, even though the physical cost of evacuation and the cost of infrastructure loss might be straight forward estimates. Woo [2008] provided one estimate of $\mathcal{C}/\mathcal{L} = 0.1$ at the very maximum if 10% of people evacuated from an area would owe their lives to that evacuation. In this case, protection would be made when $p > 0.1$.

While lives are not commonly lost due to lava flows with the notable exceptions of Laki (A.D. 1783, Iceland) and Nyiragongo (A.D. 1977 and 2002, Democratic Republic of the Congo) [Peterson and Tilling, 2000], the "protection threshold" $p > 0.1$ can still be used as an example. All Monte Carlo sub-regions have a probability of inundation ($\Pr(Lava)$) greater than 0.1 except the sub-region $\Pr(Sim < 0.1)$ (Figure 15). If action to protect is made at $\Pr(Lava > 0.1)$, then evacuations or fortifications will be made for all locations inundated by $\geq$10% of Monte Carlo simulations.

Two Bayes-factor thresholds in Figure 15 have odds against lava inundation at better than 10:1 (i.e. $\Pr(Lava < 0.1)$), "strong evidence" and "very strong evidence" against inundation. Given the hypothetical protection threshold of $p > 0.1$, the decision to protect against a lava flow should be made for a location whenever the Bayes factor supporting inundation is $> 10^{-}1$, or when there is less than "strong evidence" against inundation for that area. If the protection threshold is decided to be less than 0.1, then the evidence against inundation will have to be even stronger to support a decision to not protect an area.

It is unknown whether sub-regions defined as $\Pr(Sim < 0.1)$ should be expected to be relatively safe areas when using Monte Carlo results at future lava flow sites. To test the effectiveness of the MOLASSES algorithm used for the Tolbachik flows in other study areas, more example flows around the world will have to be tested.

5.2.3 Conclusions

Regardless of how a lava flow model is chosen, its value in forecasting lava flow hazards can be quantified using Bayesian statistics. Whether the model is bad or good, if it can be compared against a real life lava flow such as the 2012-3 Tolbachik flow, the two posteriors discussed in this paper can be calculated.

By calculating $\Pr(\neg Lava|\neg Sim)$, one may update their belief of safety based on a negative, or not-hit, result from the simulation. By calculating $\Pr(Lava|Sim)$, one may update their belief of destruction by lava based on a positive hit result from the simulation. These two tools are potentially the most important metrics by which decision makers should base their faith in a given lava flow model.

By comparing the posterior values of multiple lava flow algorithms, the best lava flow algorithm can be identified. The more important posterior metric is the $\Pr(\neg Lava|\neg Sim)$, as a low value would indicate more false negatives, resulting in more unpredicted destruction. $\Pr(Lava|Sim)$ is important but less so as low values indicate more false positives, which results in greater levels of unpredicted non-destruction. While these are a trade off, selecting the best model to decrease economic loss can be achieved by taking a weighted average of the two statistics. The weight given to each would depend on potential social or economic loss from evacuation or destruction of property. In this report, the best Pulse Volume was identified to be 4387 m^3 over TanDEM-X data in this area.

The two Bayesian posteriors are an improvement over model sensitivity and specificity as they provide a probability estimate that a simulated result is correct. By incorporating model uncertainty and performing a Monte Carlo for the MOLASSES lava flow algorithm, $\Pr(Sim)$ can be estimated for each given location, improving the usefulness of Bayesian statistics in hazard analysis.

6 Conclusions

The MOLASSES framework used in this project provides a modular set of functions that can be interchanged to fundamentally alter the flow simulation algorithm. While only 8 different algorithms, which focused on simple variations in the transition function of Cellular Automata codes, were explored, different modules can feasibly be modified to change other flow characteristics (e.g. vent geometry, mass flux at source locations, temperature dependent flow).

All CA and CA-like codes, including SCIARA [Crisci et al., 2004], MAGFLOW [Del Negro et al., 2008], ELFM [Damiani et al., 2006], LavaPL [Connor et al., 2012], and MOLASSES share similarities. For instance, lava inundation is a binary condition for each location on a grid and source locations are set at defined grid cell coordinates. This enables a common set of benchmarks to be defined, which can be used to test the validity of all CA codes.

Three validation levels have been identified and different benchmark tests have been given which apply to each level. After a code is successfully demonstrated to conserve mass, the first validation level ensures that the code does not change when parameters are changed in a meaningless way. The example benchmark is that simulated flows should not be slope-direction dependent. The second level compares simulations to analytical or experimental results. As lava flows on large scales are similar to Bingham fluids, a flow simulator should produce a circular pattern on a flat surface. The third and final level tests simulations against true lava flows. Here, the 2012-3 Tolbachik lava flow is used as an example, with flow parameters identified using TanDEM-X analysis [Kubanek et al., 2015]. It might only make sense to validate codes at higher validation levels once they have been sufficiently developed to be validated at lower levels.

Two common fitness tests for lava flows are the Jaccard coefficient and model sensitivity. Model sensitivity gives the percentage of a mapped lava flow that a simulation successfully forecasts. A possible alternative to these tests is to use Bayesian posterior probabilities to evaluate the positive and negative predictive values of the simulation. These posteriors ($\Pr(Lava|Sim)$ and $\Pr(\neg Lava|\neg Sim)$ give the likelihood of flow inundation or not inundation given the simulation results.

Once lava simulators have been successfully validated through benchmarking, these Bayesian statistics can be used to improve model input parameters and evaluate model uncertainty by incorporating uncertainty inherent in input parameters. It is also possible that the two posterior predictive values can be used to aid in decision making processes associated with active lava flow crises.

7 Data Statement

This code is available for free use on GitHub at the USFVolcanology page located at `https://github.com/USFvolcanology`, while the benchmarking codes can be found at `https://github.com/jarichardson/MOLASSES_benchmarking`. The MOLASSES code and the Benchmarking algorithms are kept in seperate self-contained repositories.

References

WP Aspinall, G Woo, B Voight, and PJ Baxter. Evidence-based volcanology: application to eruption crises. *Journal of Volcanology and Geothermal Research*, 128(1):273–285, 2003. doi: 10.1016/S0377-0273(03)00260-9.

Maria Vittoria Avolio, Gino Mirocle Crisci, Salvatore Di Gregorio, Rocco Rongo, William Spataro, and Giuseppe A Trunfio. Sciara $\gamma 2$: An improved cellular automata model for lava flows and applications to the 2002 etnean crisis. *Computers & Geosciences*, 32(7): 876–889, 2006.

D Barca, GM Crisci, S Di Gregorio, and F Nicoletta. Cellular automata for simulating lava flows: a method and examples of the etnean eruptions. *Transport theory and statistical physics*, 23(1-3):195–232, 1994.

Maria J Bayarri, James O Berger, Rui Paulo, Jerry Sacks, John A Cafeo, James Cavendish, Chin-Hsu Lin, and Jian Tu. A framework for validation of computer models. *Technometrics*, 49(2), 2007.

Alexander Belousov, Marina Belousova, Benjamin Edwards, Anna Volynets, and Dmitry Melnikov. Overview of the precursors and dynamics of the 2012–13 basaltic fissure eruption of tolbachik volcano, kamchatka, russia. *Journal of Volcanology and Geothermal Research*, 299:19–34, 2015.

Laura J Connor, Charles B Connor, Khachatur Meliksetian, and Ivan Savov. Probabilistic approach to modeling lava flow inundation: a lava flow hazard assessment for a nuclear facility in armenia. *Journal of Applied Volcanology*, 1(1):1–19, 2012. doi: 10.1186/2191-5040-1-3.

Gino M Crisci, Rocco Rongo, Salvatore Di Gregorio, and William Spataro. The simulation model sciara: the 1991 and 2001 lava flows at mount etna. *Journal of Volcanology and Geothermal Research*, 132(2):253–267, 2004.

GM Crisci, G Iovine, S Di Gregorio, and V Lupiano. Lava-flow hazard on the se flank of mt. etna (southern italy). *Journal of Volcanology and Geothermal Research*, 177(4):778–796, 2008.

Maria Luisa Damiani, G Groppelli, G Norini, Elisa Bertino, A Gigliuto, and Andrea Nucita. A lava flow simulation model for the development of volcanic hazard maps for mount etna (italy). *Computers & geosciences*, 32(4):512–526, 2006.

Ciro Del Negro, Luigi Fortuna, Alexis Herault, and Annamaria Vicari. Simulations of the 2004 lava flow at etna volcano using the magflow cellular automata model. *Bulletin of Volcanology*, 70(7):805–812, 2008.

LS Glaze and SM Baloga. Dem flow path prediction algorithm for geologic mass movements. *Environmental & Engineering Geoscience*, 9(3):225–240, 2003. doi: 10.2113/9.3.225.

RW Griffiths. The dynamics of lava flows. *Annual Review of Fluid Mechanics*, 32(1):477–518, 2000.

Andrew J Harris and S Rowland. Flowgo: a kinematic thermo-rheological model for lava flowing in a channel. *Bulletin of Volcanology*, 63(1):20–44, 2001.

Masataka Hidaka, Akio Goto, Susumu Umino, and Eisuke Fujita. Vtfs project: Development of the lava flow simulation code lavasim with a model for three-dimensional convection, spreading, and solidification. *Geochemistry, Geophysics, Geosystems*, 6(7), 2005.

Kazuhiro Ishihara, Masato Iguchi, and Kosuke Kamo. Numerical simulation of lava flows on some volcanoes in japan. In *Lava Flows and Domes*, pages 174–207. Springer, 1990.

Harold Jeffreys. *The theory of probability*. OUP Oxford, 1998.

Jim Kauahikaua, Sandy Margriter, Jack Lockwood, and Frank Trusdell. Applications of gis to the estimation of lava flow hazards on mauna loa volcano, hawai'i. *Mauna Loa Revealed: structure, composition, history, and hazards*, pages 315–325, 1995.

CRJ Kilburn. Lava flow and flow fields. In H. Sigurdsson, editor, *Encyclopedia of Volcanoes*, pages 331–343. Academic Press, San Diego, CA, 2000.

Julia Kubanek, Jacob A. Richardson, Sylvain J. Charbonnier, and Laura J. Connor. Lava flow mapping and volume calculations for the 2012–2013 tolbachik, kamchatka, fissure eruption using bistatic tandem-x insar. *Bulletin of Volcanology*, 77(12), 2015. doi: 10.1007/s00445-015-0989-9.

Warner Marzocchi and Gordon Woo. Probabilistic eruption forecasting and the call for an evacuation. *Geophysical Research Letters*, 34(22):n/a–n/a, 2007. ISSN 1944-8007. doi: 10.1029/2007GL031922. L22310.

Hideaki Miyamoto and Sho Sasaki. Simulating lava flows by an improved cellular automata method. *Computers & Geosciences*, 23(3):283–292, 1997.

D.W. Peterson and R.I. Tilling. Lava flow hazards. In H. Sigurdsson, editor, *Encyclopedia of Volcanoes*, pages 957–971. Academic Press, San Diego, CA, 2000.

Ernesto Rodríguez, Charles S Morris, and J Eric Belz. A global assessment of the srtm performance. *Photogrammetric Engineering & Remote Sensing*, 72(3):249–260, 2006.

Gordon Woo. Probabilistic criteria for volcano evacuation decisions. *Natural Hazards*, 45 (1):87–97, 2008. doi: 10.1007/s11069-007-9171-9.