University at Buffalo
The State University of New York

Scaling the Science:

Volcanic Hazard Analysis using HPC, Hazardous Mass Flow Modeling, Statistical Modeling and Parallel Analytics

A. K. Patra*, K. Dalbey*, M. D. Jones^, E. B. Pitman#, and E. Calder$
*Department of Mechanical and Aerospace Engineering,
^Center for Computational Research,
#Department of Mathematics,
$Department of Geology, University at Buffalo, State University of New York
{abani,kdalbey}@eng.buffalo.edu, {jonesm,pitman}@buffalo.edu

Abstract

We show that the efficient use of HPC, parallel adaptive simulations of the flow physics and statistical models based on the Bayes Linear methodologies enable a large-scale workflow for the measure of hazard probabilities in pyroclastic flows in a timely fashion (O(hours) as opposed to days/weeks required otherwise).

We use a multi-level hierarchical construct for the statistical model, and a load balancing master/server utility for allocating the mission-critical workflow tasks on large scale computing platforms for the case of Montserrat island. This methodology is sufficiently general that we anticipate application to a wide array of hazard analyses in response to critical geophysical events.

Challenges

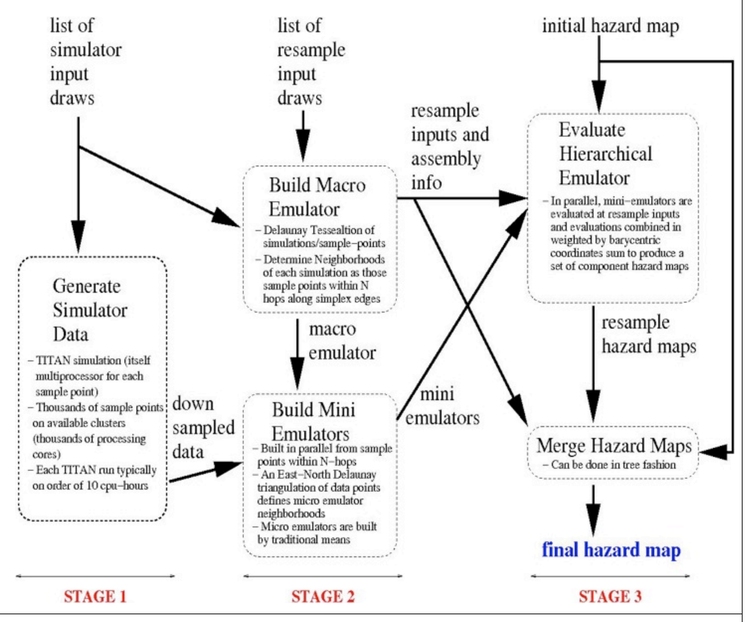
I. Probability of hazard computation at all points on hazard area requires $O(10^6)$ simulations, each of which requires O(1hour) on O(10) processors and analytics using subsets from O(1PB) distributed data!
II. Management of highly complex and dynamic workflow comprised of simulations of indeterminate length is necessary to construct hazard maps from elevation data, observations, simulation outputs and user feedback.
III. Optimize usage of available machine resources per simulation and in the overall analytics to speed up workflow.

Key Innovations

1. Use carefully constructed statistical surrogates(Bayes linear emulators) which are fast to evaluate to reduce number of full simulations required to $O(10^3)$ from $O(10^6)$.
2. Develop new methodology to parallelize construction of these emulators from multiple simulator outputs.
3. Adaptive mesh refinement, distributed data management and dynamic model based load balancing to provide efficient simulator on HPC platforms.

Approach:

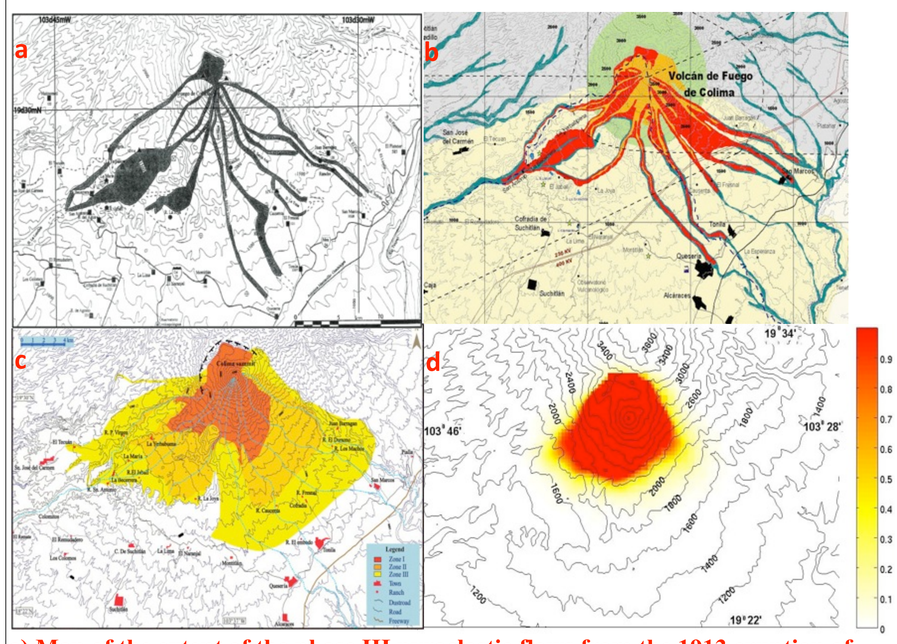
Stage 1: Evaluate an ensemble of several hundred to several thousand multiprocessor landslide simulations, dynamically assigning simulations to processors as they become available to continually **use the entire pool** of processors efficiently.
Stage 2: Create a multi-level hierarchical emulator (a statistical model) from the output of the ensemble of simulations. Its **hierarchical** nature allows the emulator's components to be constructed (and evaluated) **concurrently.** Emulator acts as a fast surrogate of the simulator.
Stage 3: Use the emulator through importance sampled Monte Carlo to compute a map of the probability that a hazard criterion will be met at hundreds of thousands (or more) of locations.



What are Geophysical Mass flows?

Geophysical mass flows include small mudflows, debris flows, pyroclastic flows, lava, and even volcanic avalanches whose volumes can exceed of 10^{11}[m^3].
•Particle sizes ranges between fine clay O(1mm) and house size boulders O(10m).
•Flow depths range from under 1 [m] up to several hundred [m].
•Run out lengths range from under 1 [km] to over 100 [km].
•There are ~1500 active volcanoes worldwide, ~60 erupt each year.

The Objective: Hazard Maps



a) Map of the extent of the phase III pyroclastic flows from the 1913 eruption of Colima (Saucedo et al 2005), b) Traditional hazard map based on field study only (Navarro & Cortez 2003), c) Hazard map based on deterministic calculations using the FLOW3D model (Saucedo et al 2005), d) Probability of flow exceeding 1 m given an event that generates a 10^7 to 10^8 [m^3] of flow volume.

The objective is to create tools that help geologists make better hazard maps through use of deterministic and stochastic modeling

Statistical Model

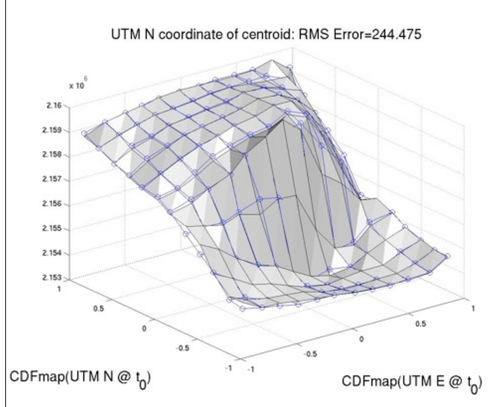
A Bayes Linear emulator consists of an approximate deterministic mean function, in this case a least squares fit, plus a Gaussian error model that is used to adjust/correct the mean function as new data is available. The equations for the emulator, its adjusted mean and its variance are

$$s_{BL}(\underline{x}) = \underline{g}(\underline{x})^T \underline{\beta} + \varepsilon(\underline{x})$$

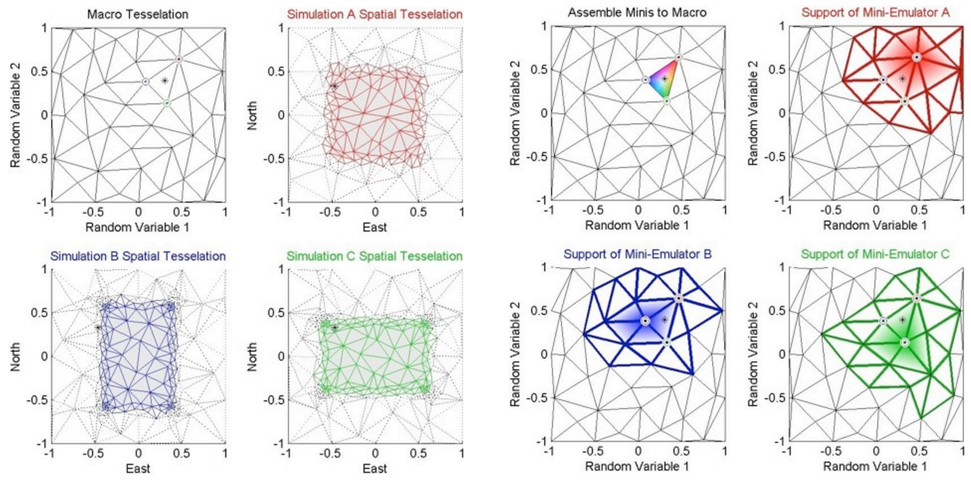
$$E_{BL}(s(\underline{x}) | \underline{s_y}) = \underline{g}(\underline{x})^T \underline{\beta} + \mathrm{Cov}(s(\underline{x}), \underline{s_y}) \mathrm{Var}(\underline{s_y})^{-1} (\underline{s_y} - \underline{g}(\underline{y})^T \underline{\beta})$$

$$\mathrm{Var}_{BL}(s(\underline{x}) | \underline{s_y}) = \sigma^2 (1 - \mathrm{Cov}(s(\underline{x}), \underline{s_y}) \mathrm{Var}(\underline{s_y})^{-1} \mathrm{Cov}(\underline{s_y}, s(\underline{x})))$$

- "s" represents the simulator output, "$\underline{x}$" is an arbitrary input, "$\underline{y}$" is a collection of inputs at which the simulator has been evaluated, "g" are the least squares basis function, β are their coefficients, and σ is the unadjusted variance, ε(x) is Gaussian model of the error.
- Equations represent the best linear (not restricted to unbiased) predictor of an unknown quantity, s(x), given the available data and choice of deterministic approximation of the mean, and form and parameters of the error model.
- Equations indicate **inherent sequential nature** of emulator.
- Hierarchical emulator is an ensemble of smaller emulators each covering a portion of the uncertain input space -- **introducing concurrency**.



A sample two-level hierarchical emulator approximating the response of the simulator for different inputs. Starting center of mass is normally distributed about summit of Colima volcano with standard deviation of 150m in East and North directions. Output is North coordinate of centroid, 600 seconds after initiation.



Upper left subplot: Delaunay tesselation of "simulation input space." Remaining subplots: East-North Delaunay triangulation of simulations A, B, and C output data. Gray rectangles indicate regions with flow, colored triangles cover flow. Black asterisk indicates a resample point.

Upper left subplot: color indicates partition of unity weight given to mini-emulators A, B, and C during assembly. Remaining subplots: shading indicates weight given to mini-emulators A, B, and C, colored lines indicate simulations used to construct mini-emulators.

Physics Model: TITAN

TITAN solves the Savage-Hutter equations for depth-averaged (shallow water type) granular flows. It has the following features:

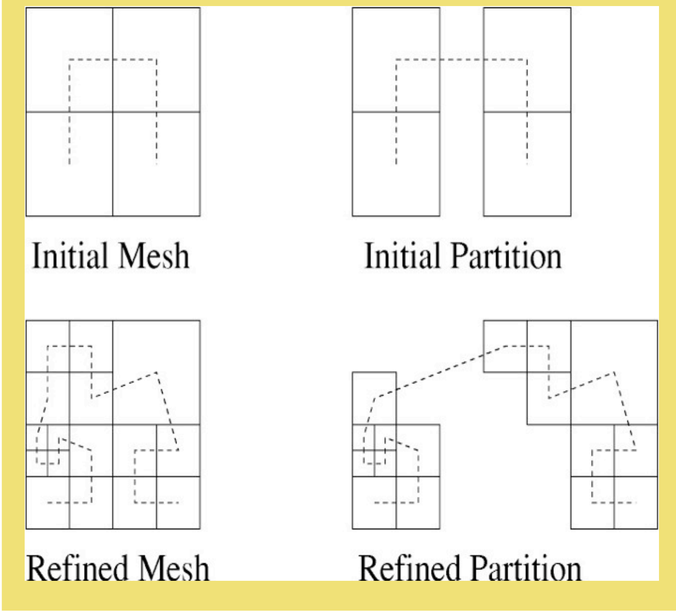
- TITAN is a high order slope-limiting upwinding two dimensional Godunov solver and uses second order accurate predictor-corrector time-stepping,
- Savage-Hutter equations lose strict hyperbolicity near the front (where h=0); this induces a non-physical "thin-layer" spreading problem in their numerical solution,
- TITAN mitigates the "thin-layer" numerical difficultiesa mesh-adaptation scheme which ensures that flow at the front will enter a "buffer layer" of maximally refined grid cells,

$$\frac{\partial h}{\partial t}+\frac{\partial hv_x}{\partial x}+\frac{\partial hv_y}{\partial y}=0$$

$$\frac{\partial hv_x}{\partial t}+\frac{\partial (hv_x^2+.5\times k_{ap}g_z h^2)}{\partial x}+\frac{\partial hv_y v_x}{\partial y}=g_x h-\mathrm{sgn}(v_x)\times\left[g_z+\frac{1}{\kappa_x}\times v_x^2\right]\times h\times\tan(\phi_{bed})-\mathrm{sgn}\left(\frac{\partial v_x}{\partial y}\right)\times\frac{1}{2}\times hk_{ap}\frac{\partial hg_z}{\partial y}\sin(\phi_{int})$$

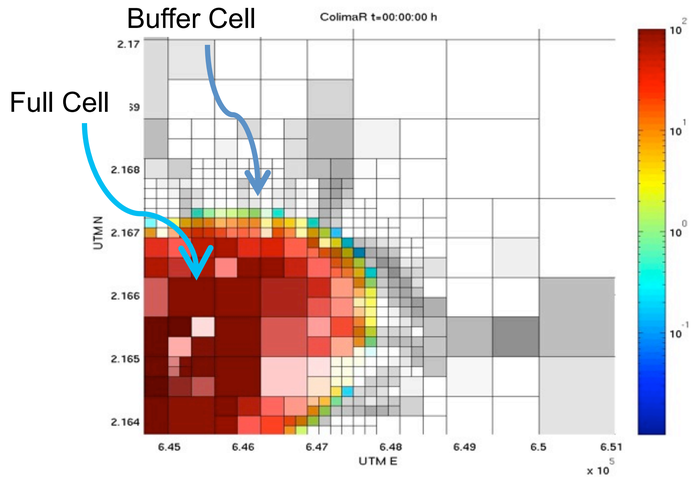
$$\frac{\partial hv_y}{\partial t}+\frac{\partial hv_y v_x}{\partial x}+\frac{\partial (hv_y^2+.5\times k_{ap}g_z h^2)}{\partial y}=g_y h-\mathrm{sgn}(v_y)\times\left[g_z+\frac{1}{\kappa_y}\times v_y^2\right]\times h\times\tan(\phi_{bed})-\mathrm{sgn}\left(\frac{\partial v_y}{\partial x}\right)\times\frac{1}{2}\times hk_{ap}\frac{\partial hg_z}{\partial x}\sin(\phi_{int})$$

- Parallel adaptive solution uses space filling curve based dynamic data management system.
- Use Real topography from integrated Geographic Information Systems
- Code is open source and works on many platforms including PCs and large scale HPC platforms - code available from http://www.gmfg.buffalo.edu



Data Driven Model Based Dynamic Load Balancing

- h-adaptive (refinement &unrefinement), gridding strategy poses a special challenge for the parallel solver,
- Dynamic space filling curve (SFC)-based load-balancing algorithm essentially divides a weighted line into segments with equal weight;
- Weights?
- 3 types of cells -- empty, full, buffer layer at front →3 different weights w_e, w_f, w_b.
  - Heuristic -- works ok for simple machines
  - Performance Model Based: goal is to minimize communication time
  - –Collect timing data for all MPI calls and total wall clock
  - –Use previous 100 time steps data and least squares model to obtain weights that minimize MPI time



$$t_{c,i}\propto w_e N_{e,i}+w_f N_{f,i}+w_b N_{b,i}$$

$$t_t-t_{w,i}=w_e N_{e,i}+w_f N_{f,i}+w_b N_{b,i}$$

$$\underline{(t_t-t_w)}=\underline{\underline{A}}\,\underline{w}\qquad \underline{w}^T=\begin{bmatrix}w_e & w_f & w_b\end{bmatrix}$$

$$\underline{w}=-\left(\underline{\underline{A}}^T\underline{\underline{A}}\right)^{-1}\left(\underline{\underline{A}}^T\underline{(t_t-t_w)}\right)$$

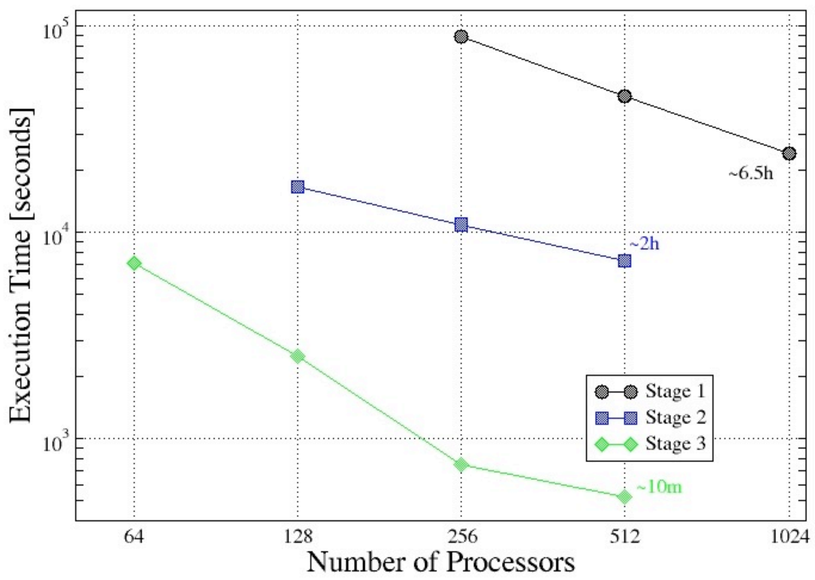
$N_{e,i}$ = # of empty cells on processor I, $N_{f,i}$ = # of full cells on processor i

$N_{b,i}$ = # of buffer cells on processor I, t_c=compute time,

t_w=wait (MPI) time, $t_c+t_w=t_t$=total time =constant across all processors

Workflow Strategy

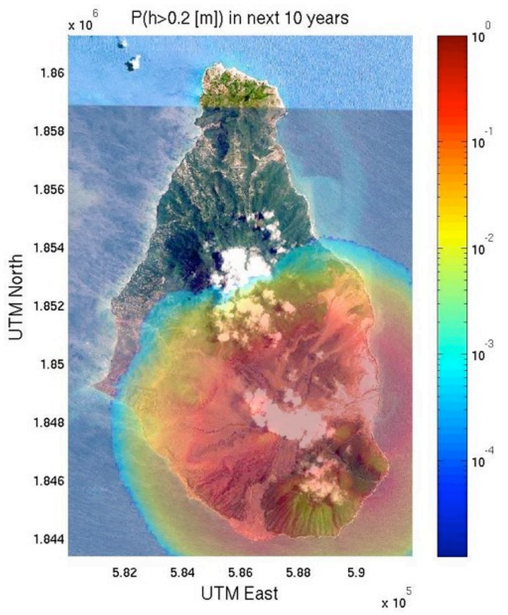
- Each stage has a similar parallelization strategy – master/worker daemon to allocate tasks to available CPUs
- I/O contention can be a serious issue (hundred of files per processor, tens of GB), so data is managed locally first, then critical inter-stage files are put on fastest available shared filesystem
- TITAN simulations (Stage 1) scale well, but still require more than 6h on 1024 processors (for only 2048 initial simulations)
- Emulator evaluation (Stage 3) is very fast, near real-time responsiveness for 512 available processors
- Principal objective achieved: simulator + emulator strategy provides very fast surrogate for pure direct simulation

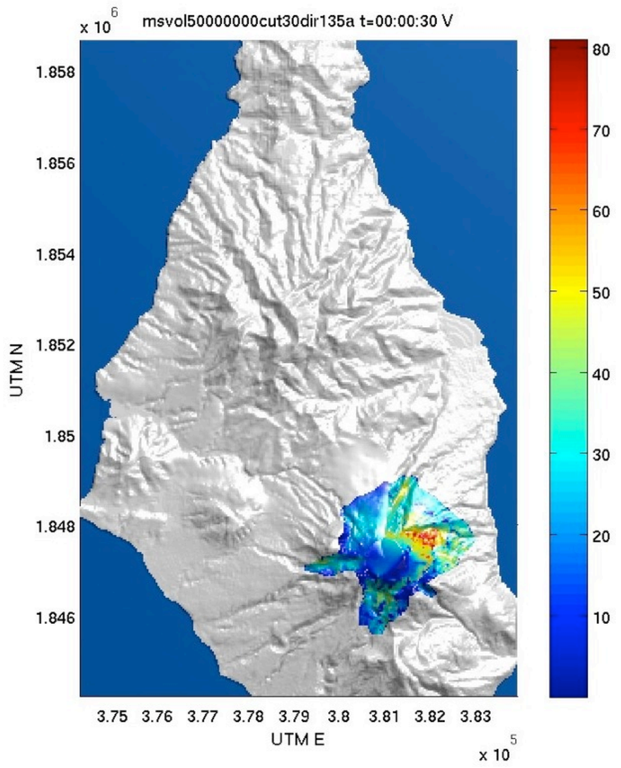


Performance speedup of three stages of the hazard map workflow: Stage 1 is generation of direct simulation inputs, Stage 2 is emulator construction, and Stage 3 is emulator evaluation (only Stage 3 needs to be redone to produce a new hazard map based on the range covered by the initial direct simulations)

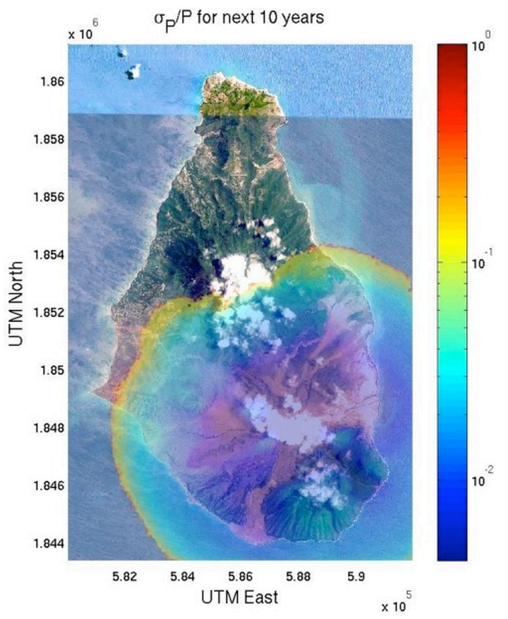
Case Study: Montserrat

- Montserrat is part of the Lesser Antilles in the Caribbean
- Soufriere Hills Volcano began erupting in 1995 destroying capital (Plymouth), forcing 2/3 of island population to flee abroad





Sample Simulation result on flow at Montserrat



Hazard maps at Montserrat computed using 2048 multi-processor TITAN simulations and 100,000 resamples of hierarchical emulator**. Total map creation time used 9 hours of wall clock on 1024 processors (classical Monte Carlo would consume ~1000 hours on 1024 processors)!!**

References

- K. Dalbey, A. K. Patra, E. B. Pitman, M. I. Bursik, and M. F. Sheridan. Input uncertainty propagation methods and hazard mapping of geophysical mass flows. J. Geophys. Res, 113, 2008.
- M. D. Jones, E. B. Pitman, A. K. Patra, K. Dalbey, and A. C. Bauer. Adaptive Simulation: Dynamic Data Driven Application in Geophysical Mass Flows. Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International, page 33b, 2005.
- M. J. Bayarri, J. O. Berger, E. S. Calder, K. Dalbey, S. Lunagomez, A. K. Patra, E. B. Pitman, E. T. Spiller, and R. L. Wolpert. Using statistical and computer models to quantify volcanic hazards. submitted to Technometrics January 25, 2008, 2008.
- 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.