Computing granular avalanches and landslides[a)]

E. Bruce Pitman and C. Camil Nichita
Department of Mathematics, The University at Buffalo, Buffalo, New York 14260

Abani Patra and Andy Bauer
Department of Mechanical and Aerospace Engineering, The University at Buffalo, Buffalo, New York 14260

Michael Sheridan and Marcus Bursik
Department of Geology, The University at Buffalo, Buffalo, New York 14260

(Received 14 June 2002; accepted 29 July 2003; published 16 October 2003)

Geophysical mass flows—debris flows, volcanic avalanches, landslides—are often initiated by volcanic activity. These flows can contain $O(10^6-10^7)$ m^3 or more of material, typically soil and rock fragments that might range from centimeters to meters in size, are typically $O(10\ m)$ deep, and can run out over distances of tens of kilometers. This vast range of scales, the rheology of the geological material under consideration, and the presence of interstitial fluid in the moving mass, all make for a complicated modeling and computing problem. Although we lack a full understanding of how mass flows are initiated, there is a growing body of computational and modeling research whose goal is to understand the flow processes, once the motion of a geologic mass of material is initiated. This paper describes one effort to develop a tool set for simulations of geophysical mass flows. We present a computing environment that incorporates topographical data in order to generate a numerical grid on which a parallel, adaptive mesh Godunov solver can simulate model systems of equations that contain no interstitial fluid. The computational solver is flexible, and can be changed to allow for more complex material models, as warranted. © *2003 American Institute of Physics.* [DOI: 10.1063/1.1614253]

I. INTRODUCTION

The risk of volcanic eruptions is a problem that public safety authorities throughout the world face several times a year.[1] Volcanic activity can ruin vast areas of productive land, destroy structures, and injure or kill the population of entire cities. The United States Geological Survey reports that globally there are approximately 50 volcanoes that erupt every year. In the 1980s, approximately 30 000 people were killed and almost a half million were forced from their homes due to volcanic activity. A 1902 volcanic gravity current from Mt. Pelee, Martinique destroyed the town of St. Pierre and killed all but one of the 29 000 inhabitants, the largest number of fatalities from a volcanic eruption in the 20th century. The 1991 eruption of Pinatubo, Philippines, impacted over 1 million people.

Hazardous activities consequent to volcanic eruptions range from passive gas emission and slow effusion of lava, to explosions accompanied by the development of a stratospheric plume with associated dense, descending volcanic gravity currents (pyroclastic flows) of red-hot ash, rock, and gas that race along the surface away from the volcano. These hot flows can also melt snow on the mountain, creating a muddy mix of ash, water, and rock. Seismic activity at a volcano can trigger the failure of an entire flank of a volcano, generating a giant debris avalanche.

Slower moving mass flows of surficial material take the form of coarse block and ash flows, debris flows, or avalanches. Some of these flows carry with them a significant quantity of water. The 1998 mud flow at Casita Volcano in Nicaragua caused thousands of deaths. Debris flows associated with the 1985 eruption of Nevado del Ruiz, Colombia, resulted in the death of 26 000 people.[2] Although scientists had developed a hazard map of the region, the people in the devastated area were unaware of the zones of safety and danger. If they had known, many could have saved themselves. Debris flows originating from severe rainstorms threaten many areas throughout the United States, Mount Rainier being one principal risk site.[3,4]

In block and ash flows, volcanic avalanches, and debris flows, particles are typically centimeter to meter sized, and the flows, sometimes as fast as hundreds of meters per second, propagate tens of kilometers. As these flows slow, the particle mass sediments out, yielding deposits that can be as much as 100 meters deep and many kilometers in length.

For agencies charged with civil protection during volcanic crises, the question they want answered is ''Should we evacuate a town or village? And if so, when?'' At present, it is sometimes possible to predict when premonitory activity might lead to a large-scale eruption. It is more difficult to predict when activity might lead to slope failure of some part of the volcano, or the generation of a debris flow. However, one can ask the following question: If a mass flow were to be initiated at a particular location, what areas are most at risk from that flow? This is the issue we address here. In particular, we describe our efforts in developing a computational

[a)]This work was originally presented at the Symposium on Multicomponent and Multiphase Fluid Dynamics, in conjunction with the 14th U.S. National Congress of Applied Mechanics. The Symposium was a celebration of Dan Joseph and his many fundamental contributions to fluid mechanics.

environment for simulations of geophysical mass flows. We present a numerical algorithm for solving the governing model equations, and describe our implementation of a parallel, adaptive grid Godunov solver to simulate flows. An important feature of this work is the incorporation of digital elevation model data into our computational framework. These ingredients allow us to simulate large flows over a realistic terrain; here we provide an example of such a simulation, at Volcan de Colima in Mexico.

II. MODELS

In this section we summarize recent efforts in modeling geophysical mass flows. These efforts began with a paper by Savage and Hutter.[5] Starting from equations of mass and momentum balance, using a Mohr–Coulomb constitutive relation, and making use of scaling arguments, they developed a one-dimensional model system similar to the shallow water equations. Savage and Hutter wrote a code to numerically solve the resulting equations, reporting some difficulties with their Eulerian-based algorithm. They also performed a series of table-top experiments, to test their predictions. Savage and Hutter extended their work to two dimensions, found a similarity solution to the governing equations, and examined flows over more general basal surfaces.[6] The effort was taken up by Iverson,[7–9] who extended the modeling to include the effects of pore fluid, and employed modern shock-capturing numerical techniques to solve the model system. Gray[10,11] also rederived the governing equations for general basal surfaces, and developed an interesting flow experiment, viz., avalanches in a rotating drum. More recently, Gray and colleagues[12] solved the one-dimensional model system by using shock-capturing combined with explicit tracking of the propagating edge of the granular mass.

We first present a heuristic derivation of a one-dimensional model system, an approach that highlights the fundamental ideas required to develop a hydraulic model from the physical system governed by Mohr–Coulomb frictional plasticity. We refer the reader to the original work by Savage and Hutter, to gain an appreciation of their approach, and the scaling arguments that guide their derivation.[5,6] We then present a more general model system appropriate for flows over two-dimensional terrains. It is of importance to recognize that the fundamental evolution equations for an incompressible Mohr–Coulomb material are linearly ill-posed.[13] That is, the basic balance laws, when linearized around a constant state, exhibit runaway growth. In spite of this ill-posedness, the Mohr–Coulomb theory is widely used for steady-state stress and flow analysis in engineering practice. Some degree of well-posedness can be recovered by allowing slight compressibility,[14,15] but the fact remains that we do not understand the basic physics of time-dependent granular flow.

Actual debris and avalanche flows are often accompanied by significant quantities of liquid mixed into the rock and soil mass. This liquid can be important in the initiation of these flows. Liquid in the interstitial matrix also changes the rheological character of the flowing material. It is not clear how to best incorporate these important effects. Because our focus here is on a unified framework for computing flows, we ignore the effects of interstitial fluid, and discuss only the flow of a single constituent material. The interested reader may consult[7] for one modeling effort that includes fluid effects.

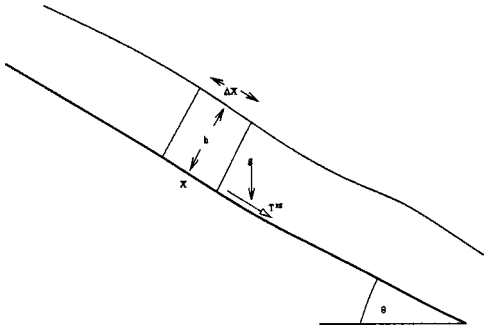

FIG. 1. A schematic diagram illustrating the average forces acting on a two-dimensional mass of granular material flowing down an inclined plane.

A. Flow in one dimension: A heuristic derivation

In this section, we illustrate the major features of a one-dimensional model of granular flow down an incline. We do so to highlight the important assumptions of such a model, and to make explicit connections between the long history of the engineering study of granular flows in bins and hoppers on the one hand, and newer applications of granular material theories to geological phenomena on the other.

In the Cartesian plane, consider a thin layer of an incompressible granular material flowing down a flat plane making an angle θ with the horizontal. Let $[x,\ b(x)]$ be a point on this plane, and let z be the direction normal to the plane; see Fig. 1. Assume a constant density of material, ρ. Consideration of mass conservation for a slice of this layer between the points $x-\Delta x/2$ and $x+\Delta x/2$ balances the time derivative of material mass, $\partial_t(\rho h \Delta x)$, and the flux of material $\rho h u$ across the edges $x+\Delta x/2$ and $x-\Delta x/2$. Taking the limit yields an equation for the evolution of the height,

$$\partial_t(\rho h)+\partial_x(\rho h u)=0.$$

The slice is subject to forces in the x and z directions due to gravity, friction, and material deformation, assumed to be characterized by Mohr–Coulomb plasticity. This Mohr–Coulomb theory, the generalization of simple sliding friction to a continuum, makes the following assumptions on material behavior.

(i) Material deforms when the total stress reaches yield, described by the condition $\|\mathrm{dev}(T)\|=\kappa\,\mathrm{tr}(T)$, where T is the 2×2 stress tensor, dev is the deviator of the tensor, $\mathrm{dev}(T)=T-1/2\mathrm{tr}(T)I$, and $\mathrm{tr}(T)$ is the trace of the stress $\mathrm{tr}(T)=\Sigma_{i=1}^{2}T^{x_i x_i}$.

(ii) As material deforms, the stress and strain-rate tensors are aligned, $\mathrm{dev}(T)=\lambda\,\mathrm{dev}(V)$; where the strain-rate, $V=-\frac{1}{2}(\partial_{x_i}u_{x_j}+\partial_{x_j}u_{x_i})$, where $\mathbf{u}=(u_1,\ u_2)$ is the velocity.

(iii) Material is rigid if stresses are below yield.

Let us now develop balance laws for $\overline{T}$, the stresses averaged over the layer depth h. If motion normal to the basal surface is negligible, the only forces in the z direction are lithostatic,

$$\partial_z \overline{T}^{zz} = -\rho g \cos(\theta).$$

Because the stress vanishes on the top surface, we find $\overline{T}^{zz} = \rho g \cos(\theta)(h-z)$, where $h(x)$ is the thickness of the layer above the point $[x,\ b(x)]$.

In the x direction, the total time rate of change of net momentum of the slice is given by $(d/dt)(\rho h u \Delta x)$, and is balanced by the derivative of the downslope stress $\partial_x \overline{T}^{xx}$, the basal friction $\overline{T}^{xz}|_b$, and the lateral component of gravity $\rho g h \sin(\theta)$.

In all its details, Mohr–Coulomb plasticity is too complex a theory to apply here. Instead we make a series of simplifying assumptions that allow us to derive a tractable set of equations. First, we assume that material is always deforming, so no region is rigid. Next, at the basal surface we assume perfect alignment of the normal stress and the shear stress, so $\overline{T}^{xz}|_b = \mu \overline{T}^{zz}|_b$. We further assume that the stresses remained aligned throughout the layer thickness; any misalignment is likely to be small if the layer is not very thick. Finally, we assume that the xx stress and the zz stress are proportional throughout the layer, $\overline{T}^{xx} = K \overline{T}^{zz}$; here K is taken to be the earth pressure coefficient, a classical factor that is widely used[16] and whose origins extend back to Rankine.[17] In essence, these assumptions replace the functional relation between stress and strain rate (and the factor λ) of plasticity by a state and space-dependent proportionality "constant" together with fixed axes of alignment, a significant simplification of the constitutive theory.

Bringing all of these terms together, and using the hydrostatic relation for the normal stress (and thus for all the stresses), the balance law reads as

$$\partial_t(\rho h u) + \partial_x(\rho h u^2 + \tfrac{1}{2}\beta \rho g h^2) = \rho g h \sin(\theta) - \mathrm{sgn}(u)\cos(\theta)\tan(\delta)\rho g h. \tag{1}$$

Here, the coefficient $\beta = K\cos(\theta)$.

The earth pressure coefficient K may not be constant, but depends on whether the local downslope and cross-slope flows are expanding or contracting (i.e., in the active or passive states, respectively). That is, K may depend on whether $\partial_x u > 0$ or $\partial_x u < 0$ at a particular location. The next section provides a specific definition and a more complete discussion of K.

We note that a fuller accounting for shearing stresses and slope changes would introduce an additional source term proportional to u^2 [see Eq. (2)]. See Refs. 8 and 11 for a derivation that includes this term.

The general approach suggested by this derivation, and the specific alignment and friction assumptions, are similar to the theory of bin loads due to Janssen,[18] see Ref. 19 for a development of the long history of related applications. The paper[20] compares the Janssen assumption with a full numerical solution of the governing equations in a circular bin. The Janssen approach is traditionally invoked only under static conditions. A central assumption of the Savage–Hutter theory is to apply this idea under *dynamic* conditions.

To better appreciate the relative sizes of terms in this model system, the equations should be scaled in both dependent and independent variables. Clearly, overall stresses can be scaled by the lithostatic force $\rho g H \cos(\theta)$, where H is a characteristic thickness of the flowing layer. This scaling removes the density as a parameter, and clears some of the trigonometric functions. But more important is a scaling of the independent variables. Scale x by L, a characteristic downslope length, z by H, and t by $\sqrt{L/g}$, and make the long wave assumption, namely $H/L = \epsilon \ll 1$. With these scalings, the equations as presented above are modified by the introduction of ϵ, modifying the pressure-like term $\beta \rho g h^2$. See Eq. (2).

Similar to the shallow water equations in structure, this system of equations is strictly hyperbolic and genuinely nonlinear away from the "vacuum state" where $h = 0$; the characteristic speeds for the system are $\lambda_\pm = u \pm \sqrt{\beta h}$.

B. Two-dimensional model equations

The approach sketched above, or a more formal derivation of model equations using scaling arguments, can be extended to flow over a two-dimensional surface.[6,8,11] As recognized by Iverson and Denlinger,[8] for the model system to apply to flow over terrain, there can be no preferential direction, as is present in earlier derivations. The governing equations must possess rotational invariance. Rather than rederive the governing equations, we simply present the governing thin layer equations as in Ref. 9, except we ignore all terms related to the presence of fluid in the mixture. Depth averaging is applied to an arbitrary element in a local coordinate system that has the Oz axis directed normal to the element surface, and Oxy the plane tangent to the terrain locally. The flowing layer has thickness h, and the velocity field is u, v in the downslope and cross-slope directions, respectively,

$$\partial_t h + \partial_x(hu) + \partial_y(hv) = 0, \tag{2}$$

$$\partial_t(hu) + \partial_x\left(hu^2 + \frac{\beta}{2} g_z h^2\right) + \partial_y(huv) + \mathrm{sgn}(\partial_y u)\,\partial_y\left(\frac{\beta}{2}\sin(\phi) h^2 g_z\right) = h g_x - \mathrm{sgn}(u) h \tan(\delta)\left[g_z + \frac{u^2}{r_x}\right],$$

$$\partial_t(hv) + \partial_x(huv) + \mathrm{sgn}(\partial_x v)\,\partial_x\left(\frac{\beta}{2}\sin(\phi) h^2 g_z\right) + \partial_y\left(hv^2 + \frac{\beta}{2} g_z h^2\right) = h g_y - \mathrm{sgn}(v) h \tan(\delta)\left[g_z + \frac{v^2}{r_y}\right].$$

Here, $\beta = \epsilon K$ is small, r_x, r_y denote the radius of curvature of the local basal surface, and g_x, g_y, g_z gives the local direction of gravity. In the computational algorithm detailed

below, one can approximate the (inverse of the) radius of curvature by derivatives of the basal slope, e.g., $1/r_x = \partial_x \theta_x$, where θ_x is the local bed slope.

Away from the vacuum state $h=0$, this system is again hyperbolic, with two nonlinear wave families similar to the one-dimensional case, and a linear family whose wave speed is the convection speed.

The earth pressure coefficient K is in the active or passive state, depending on whether the downslope and cross-slope flows are expanding or contracting. Gray[11] defines four values of K, depending on whether $\partial_x u$ and $\partial_y v$ are positive or negative. We follow Iverson,[8] who defines two values for K:

$$K_{\text{act/pas}} = 2\,\frac{1 \pm [1 - \cos^2(\phi)[1 + \tan^2(\delta)]]^{1/2}}{\cos^2(\phi)} - 1,$$

where the + (passive state) applies when flow is converging—that is, if $\partial_x u + \partial_y v < 0$—and the − (active state) applies if $\partial_x u + \partial_y v > 0$.

III. COMPUTING FLOW ALONG A TWO-DIMENSIONAL SURFACE

In this section, we briefly describe TITAN2D, a parallel, adaptive grid, shock capturing method to solve the governing equations (3). We begin with a discussion of the integration of digital elevation model data—a description of the local topography—into our solver. The synthesis of these computational techniques makes possible the solution of mass flows over a realistic terrain.

A. Digital elevation data

A principal feature of our code is the incorporation of topographical data into our simulation and grid structure. We have written a preprocessing routine in which digital elevation data are imported. These data define a two-dimensional spatial box in which the simulation will occur. The raw data provide elevations at specified locations. By using these data, and interpolating between data points where necessary, a rectangular, Cartesian mesh is created. This mesh is then indexed in a manner consistent with our computational solver. The elevations provided on this mesh are then used to create surface normals and tangents, ingredients in the governing PDEs. Finally, the grid data are written out for use, together with simulation ouput, in post-computation visualization.

The digital elevation data may be obtained from a number of geographic information system (GIS) sources. We have implemented a version that imports GRASS data (Geographic Resource Analysis Support System), which is then georectified and coded into a grid, in a manner similar to the GRID module of ARC/INFO.[21] Although we wish to facilitate the use of GIS data standards in our simulation environment, commercial GIS software is not efficient at data retrieval. In addition, hierarchical and variable resolution elevation data are not readily available in commercial packages. Our preprocessing module addresses at least some of these difficulties.

Currently our GIS data provide a uniform degree of resolution. Depending on the specific site under consideration, a typical coarse grid provides blocks of about 100 m $\times$100 m in size with a $\pm$30 m vertical accuracy, while a fine grid has blocks of 5 m$\times$5 m, with $\pm$1 m vertical accuracy. Of course, it is possible to have elevation data of differing fidelity in a region of interest. In such a case, the code must query the data source for the most accurate information available, and the computational mesh must be generated appropriately. Building the software apparatus to query the GIS data during mesh refinement would greatly improve the overall accuracy of our simulations.

B. Finite volume Godunov solver

To solve this hyperbolic system of equations (2), we use a parallel, adaptive mesh, Godunov solver. The basic ingredient in the method is an approximate Riemann solver. We have coded a solver originally due to Davis,[22] based on the ideas of Harten, Lax and vanLeer.[23] In brief, the dependent variables are considered as cell averages, and their values are advanced by a predictor–corrector method; the source terms are included in these updates, and no splitting—neither for the source nor the multiple dimensions—is necessary. Slope limiting is used to prevent unphysical oscillations. The Davis approximate Riemann solver is a centered scheme, akin to an approach introduced by Rusanov.[24]

Consider, then, a hyperbolic system written as

$$\partial_t U + \boldsymbol{\nabla}_x f(U) + \boldsymbol{\nabla}_y g(U) = s(U),$$

or, if A and B are the Jacobians of f and g, respectively, as

$$\partial_t U + A\boldsymbol{\nabla}_x U + B\boldsymbol{\nabla}_y U = s(U).$$

Given U_{ij}^n, the (i,j) cell average of U at time $n\Delta t$, the mid-time predictor is

$$U_{ij}^{n+1/2} = U_{ij}^n - \frac{\Delta t}{2} A_{ij}^n \Delta_x U_{ij}^n - \frac{\Delta t}{2} B_{ij}^n \Delta_y U_{ij}^n + \frac{\Delta t}{2} s_{ij}^n.$$

In this formula, $\Delta_x U$ and $\Delta_y U$ are the limited slopes for U in the x and y directions, respectively.

Now at, say, cell edge $i+1/2,j$, there are two values that U may have, namely, the left state $U^l_{i+(1/2)j} = U_{ij}^{n+1/2} + (\Delta x/2)\Delta_x U_{ij}^n$ or the right state $U^r_{i+(1/2)j} = U_{i+1j}^{n+1/2} - (\Delta x/2)\Delta_x U_{i+1j}^n$. To resolve this multivaluedness, an approximate Riemann solver generates a numerical flux $F(U^l, U^r)$ depending on these left and right states and the physical flux f. The interested reader may consult Refs. 25 and 26, among other sources, for a guide to the vast literature of Riemann solvers. We follow the early work of Davis,[22] who generates a solver based on a single intermediate state, produced by the examination of the fastest and slowest wave speeds propagating from the local states:

$$F(U^l, U^r) = \frac{1}{2}[f(U^r) + f(U^l)] - \frac{\alpha}{2}(U^r - U^l).$$

In this formula, α is an upper bound on the magnitude of the characteristic speeds of all waves (normal to the cell edge under consideration) evaluated at both the l and the r states. A related Riemann solver (HLL or its HLLE extension[26])

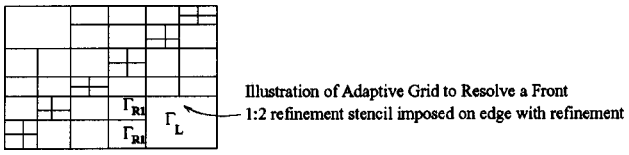

FIG. 2. Grid transition with a conservative distribution of the flux across the interface Γ_L and $\Gamma_{R1} \cup \Gamma_{R2}$.

uses two intermediate states—in essence, a complete Riemann solution for our system. We have found there to be little difference between these two solvers in our simulations, provided the computational grid is sufficiently fine.

Finally, a conservative updated of U is computed as

$$U_{ij}^{n+1} = U_{ij}^{n} - \frac{\Delta t}{\Delta x}[F(U_{i+(1/2)j}^{l}, U_{i+(1/2)j}^{r}) - F(U_{i-(1/2)j}^{l}, U_{i-(1/2)j}^{r})] - \frac{\Delta t}{\Delta y}[G(U_{ij+(1/2)}^{b}, U_{ij+(1/2)}^{t}) - G(U_{ij-(1/2)}^{b}, U_{ij-(1/2)}^{t})] + \Delta t s_{ij}^{n+(1/2)}. \quad (3)$$

Here U^b and U^t are bottom and top states (for y fluxes) analogous to U^l and U^r. Although the Davis method (and the more general HLL solver) is not as accurate as other solvers, its ease of use for systems with sources and for systems in several spatial dimensions, and its small operation count, recommend its use for our model equations.

We note that this approach avoids a splitting of the source terms from the propagation terms. Splitting often creates difficulties for quasisteady flows. In particular, unless special precautions are taken, many hyperbolic solvers (including the Davis and HLL solvers) introduce dissipation that can destroy special time-independent solutions of

$$\nabla_x f(U) + \nabla_y g(U) = s(U).$$

See Refs. 25 and 26 for details.

C. Adaptive gridding

Since the work of Berger and Collela,[27] the advantages of adaptive gridding for an accurate resolution of solutions to hyperbolic partial differential equations have been recognized. Together with parallel computing to enable the use of very fine grids, these methods provide the opportunity for simulations of unmatched fidelity.

Here we have implemented a simple adaptive grid structure (see Fig. 2) that refines cells based on indicators derived from the solution at the previous time step. Perhaps the simplest indicator to consider is "Is $h>0$?" That is, all cells containing material are refined. In experiments with this simple indicator, we may also refine all cells immediately adjacent to cells containing material. In this way, a one-cell thick band of refined but empty ($h=0$) cells surrounds the refined cells containing material. This additional band of refined cells ensures that no spurious waves are generated as material passes across different grids. Alternatively, enforcing precise conservation at edges where refined and unrefined grids meet also provides a highly accurate computation.

We have also experimented with a simple scaled L^2 norm of the flux around the boundary of each cell as the refinement indicator. That is, the refinement indicator for the kth cell is $\eta_k = 1/d_k \oint_{\partial\Omega_k} |F|^2 dx$, where d_k is the diameter of the cell and $\partial\Omega_k$ is its boundary. The cells to be refined are those for which η_k is the largest p% of all cells; of course, p is adjusted by the user.

In a similar way, if indicators suggest that a coarser discretization will not adversely affect the solution quality, we remove refined cells. Thus, we are able to maintain the solution quality and track special features, while not making the computation prohibitively expensive.

We also monitor the change of the pile height on every time step: $h(t_n) - h(t_{n-1})$; when this change is below a user defined threshold, we unrefine the grid locally.

Both the refinement and unrefinement indicators are heuristic, and based on experience and physical intuition.

We must also define the frequency of grid adaption. In the calculations reported here, we examine the indicators every two time steps to decide whether or not to refine or unrefine. Again, this adaption frequency is heuristic. We are investigating more sophisticated strategies, based on estimates of local error.

In Fig. 6 of Sec. IV, a grid adapted to a simulation over a realistic terrain is shown.

D. Parallelization

Parallelization of adaptively refined grids has been addressed by several researchers over the last several years; see, for example, Flaherty *et al.*,[28] Carter and Stewart,[29] and Laszloffy, Patra, and Long.[30] To take full advantage of multiprocessor computing, the critical issues are the number and location of cells created and deleted during grid refinement, load balance, and the efficiency of storage and data access.

For the model system here, we extend the approach described in our earlier work on adaptive steady-state finite element codes.[30] The primary modifications to this earlier work that have been necessary to accommodate the time-dependent hyperbolic nature of our model equations are (1) the addition of modules to enable the unrefinement of cells; and (2) the addition of data storage for a layer of "ghost cells" on each subdomain.

These "ghost cells" are a layer of cell data for all cells adjacent to the line (in two dimensions) that partitions the grid among some pair of processors. The data of a cell adjacent to the partition and belonging to one processor are replicated on the other processor. This replication enables us to perform the computations necessary for the Godunov scheme for every cell, without explicit communication. However, the use of ghost cells requires us to synchronize the data among the processors at the end of each time step.

E. Data management and load balancing

Although we use only regular Cartesian grids in the simulations reported here, our data management scheme is imported from FEM computations, where unstructured grids

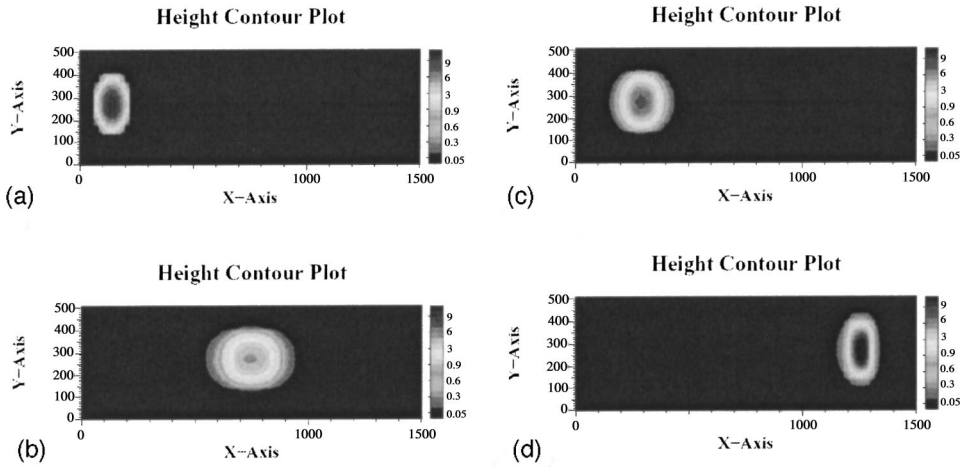

FIG. 3. Granular avalanche down an inclined plane. Contour plots show a paraboloid shape, which can be shown to be a similarity solution of the governing system. The plane makes an angle of 45° until 1000 m, at which point the surface rapidly flattens to become horizontal. The internal friction angle of the flowing mass is 30°, and the bed friction is 20°.

often arise. Thus our design allows for general grids, and our data structure are not simple arrays. Because all data access operations involve some kind of search, this procedure must be fast and efficient.

Our data storage model is a distributed hash table. Data are indexed by an ordering along a space filling curve (SFC), and are then stored in the table.[30] Each cell is mapped to a unique key using its location along the SFC passing through its centroid. This key also provides a unique identifier that is easily generated (typically it is generated directly from geometric coordinates by using bit manipulations). To obtain a decomposition of the problem, we introduce a partitioning of this key space. This induces a distribution of the data sets, i.e., a decomposition of the problem for distributed memory computing. When the grid changes and new cells are introduced, a redistribution of cells among the processors to maintain load balance is achieved by adjusting this key space partition.

F. Parallel solver

A parallel solution of the model partial differential equations of Sec. II is easily implemented within the infrastructure just described. After a decomposition of the domain is obtained using the key space partitioning, we introduce one layer of ghost cells on each processor. Each cell maintains a list of the keys of its neighbors. These features, together with the finite range of influence (of any individual cell) that characterizes a hyperbolic problem, make access to on-processor and off-processor data for the Godunov solver relatively simple. By adjusting the key space partition after grid refinement/unrefinement stages, good load balance is maintained.

IV. RESULTS

We demonstrate our computational solver by examining two test cases. The first flow is a mass moving down an inclined plane. For this case, topography is not an issue, and the simulation probes the accuracy of the numerical solver. The second case is flow down Volcan de Colima in Mexico, and the topographical features of the region largely determine the distribution of the mass flow and deposit. Additional comparisons may be found in Refs. 31 and 32.

A. Flow down an inclined plane

This is a simple test of the numerical methodology described here. Laboratory experiments using this type of geometry have been described by Refs. 5, 9, and 11, among others. We show in Fig. 3 flow down an inclined plane at a 45° angle. At 1000 m, the surface becomes horizontal. Savage and Hutter[5] found that a parabolic distribution of material is a similarity solution to the governing equations in one dimension, and Ref. 6 derives an analogous solution for two-dimensional flows. This result is borne out in our simulations. Figure 3 shows contour plots of the flowing hump, and Fig. 4 gives a cross section through the middle of the mass, approximately halfway down the incline. After the hump reaches the flat portion of the surface, flow quickly ceases. For this run, a free-flowing material was used, with an internal friction angle of 30°, and a bed friction of 20°.

B. Realistic terrain

Our efforts in this project are motivated by the desire to simulate debris and avalanche flows over realistic terrain. To

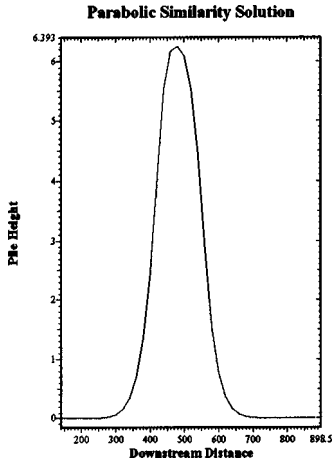

FIG. 4. A cross-sectional profile of the sliding granular mass approximately halfway down the inclined plane.

this end, we have selected a small set of volcano sites at which we have detailed information about historical mass flow events, information either from real-time reports of the event or from geological field work and reconstruction. Our simulations, then, will attempt to reproduce these flows with a minimum of parameter adjustment. To this end, the model introduced in Sec. II is well-suited—there are only two parameters to choose: the internal and basal friction angles. Here we report on a simulated flow at Volcan de Colima in Mexico.

In 1991, a block and ash flow of about $\sim 10^6$ m^3 initiated from the SW face of a lava dome located at the summit. We do not know the exact geometry of the failure surface. The TITAN2D computation begins by loading topographical data—recent digital elevation data for Colima and its surroundings. For the simulation reported here, this elevation data are gridded by cells 60 m$\times$60 m, with a vertical accuracy that, in some locations, is only $\pm$40 m.

The initial mass is assumed to be of a paraboloid shape, 10^6 m^3 in volume, located near the top of the cone. We assume internal and bed friction angles of 30 and 15, respectively. This bed friction value is lower than what is often measured for geologic materials, but the initial mass flowed easily upon release; simulations with a more realistic bed friction value of 25° did not flow as rapidly as desired.

The paraboloid shape exaggerates the initial inclination of the flowing material, and makes the very early time course of our simulation highly questionable. The flow a little later appears more as is to be expected. The contour maps of Fig. 5 show the flow simulation at two times. The map on top shows the mass as it gathers momentum and begins to elongate. The map on the bottom is somewhat later; the mass has consolidated again, as it encounters the hillock shown in the terrain map. Interestingly the flowing mass does split along a ridge between barancas, similar to the real flow. The simulated flow does not channelize in quite the same way as the 1991 flow did, and there are real differences between our computations and the 1991 event. Nonetheless, the qualitative features and general characteristics match reports of this avalanche.[33]

In Fig. 6, the grid corresponding to the flow times of Fig. 5 is shown. Here one can see the refinement of the computational grid as it follows the flowing mass.

(*Aside*: In this Colima simulation and many others, the digital elevation data available incorporate the deposit of and erosion from the very flows we wish to simulate. There are few sites where pre- and post-flow elevation data are available. We do not know how much our simulations are impacted by such topographical inaccuracies.)

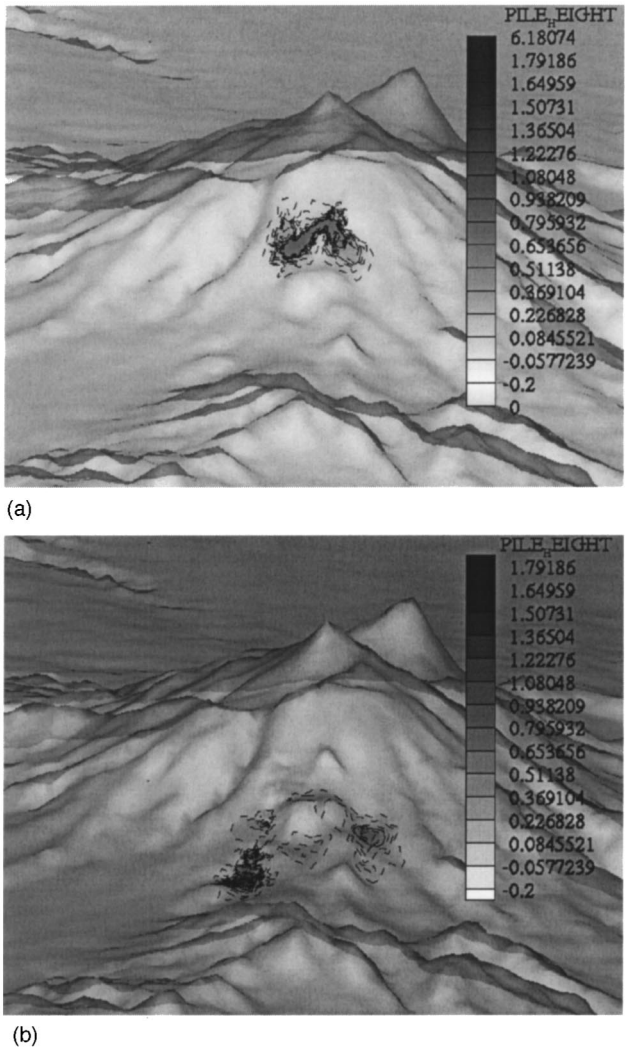

FIG. 5. The spreading of an avalanche, modeled using the thin layer system (3) and terrain data from Volcan de Colima in Mexico. The contour maps give the flow depth at two different times. On the top, the mass is accelerating and elongating as it flows downhill. On the bottom, the mass encounters a small hillock and consolidates as it spreads around the obstacle. The dashes represent contours of pile height lower than those of the darker shaded mass.

V. DISCUSSION

We now have in place a computational environment for simulating the system of equations modeling single-phase geophysical mass flows, a computational environment that integrates accurate topographical data into a parallel, adaptive mesh Godunov solver. Simulation results are qualitatively correct, but detailed experimental work, especially large-scale field work against which to compare simulations, is still required. Although the Mohr–Coulomb theory at the heart of the physical modeling presented here requires only a small number of physical parameters (internal and basal friction angles), these parameters are only loosely defined, and the measurements of these quantities are crude.

On the one hand, our model system is flexible enough to allow several candidate constitutive relations (which may not suffer the ill-posedness of the Mohr–Coulomb theory) to be used. On the other hand, any ''thin layer'' model of flow, by necessity, hides important information through the averaging process. An open problem is to recover some of that hidden information—for example, to understand the internal dynamics of recirculation—without recourse, to a direct simulation of the complete Mohr–Coulomb model equations. A related question is the following: Does the specific constitutive theory employed in such a thin layer model qualitatively change simulation results?

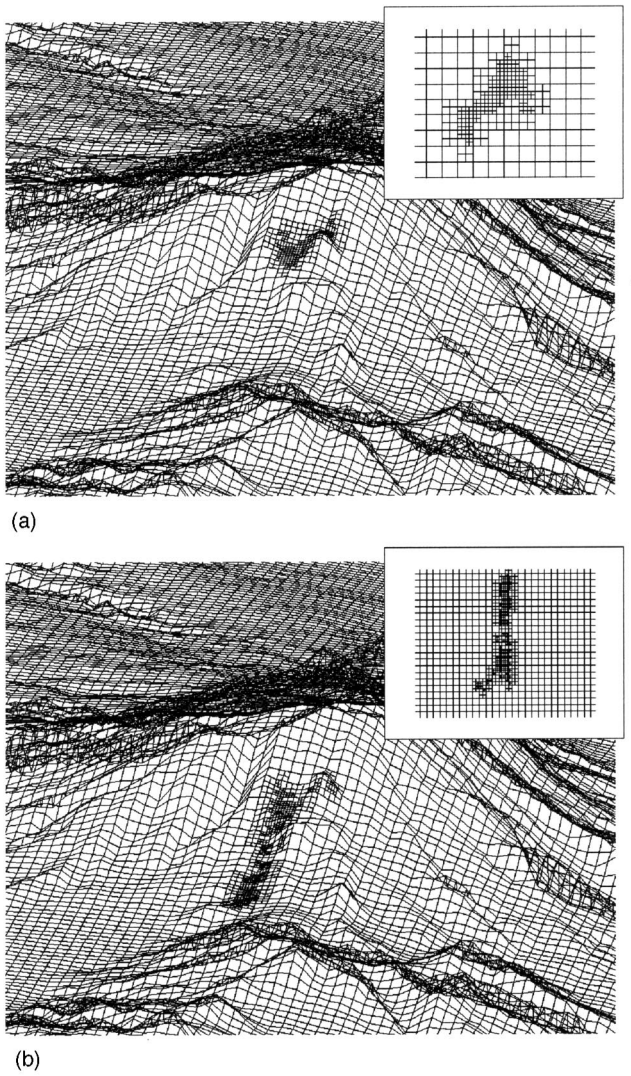

FIG. 6. The computational grid, adapted to the moving mass. The two figures show the refined grid corresponding to the two flow times of the previous figure. The inset is a blow-up of the region immediately around the flowing mass, showing up to three levels of refinement.

The numerical method described here, and similar approaches by Refs. 9 and 11, are formally second-order accurate (though perhaps with a first-order splitting of the source terms). The sheer size of the regions on which we wish to simulate flow, however, normally necessitates large grid cells. Our adaptive gridding and parallel framework gives us the capacity to perform computations on reasonably sized grid cells. Our experience shows that a grid with sufficiently small cells is necessary for reasonable simulations. At the same time, elevation data must be accurate enough to allow these fine grids to be faithful to the actual topography. These topographical data are available down to a $5\,\mathrm{m}\times 5\,\mathrm{m}$ scale with a ± 1 m vertical accuracy. We are investigating a strategy for computational mesh refinement coupled to a hierarchy of refined topographical mappings. This marriage is especially important in circumstances where very accurate elevation data are available only in a portion of the simulation area.

The numerical solver for the thin layer equations suffers from a modeling defect easily appreciated by considering finite size computational cells. Equations (3) are derived by depth averaging, referred to as a basal surface that conforms (as best as possible) to the local topography. Computational cells can be thought of as being locally tangent to this topography. Thus, for two adjacent cells, the numerical fluxes across cell edges are obtained by depth averaging in slightly different directions. Unfortunately, this discrepancy does not disappear as the cell sizes shrink to zero. In spite of the shortcomings of the modeling and the computational approaches, the thin layer equations appear to give good results—even when they may be supposed not to hold.

Another concern for applications to real systems is the bulking of material during flow due to erosion. It is estimated that, in large debris flows, up to half—or more—of the final deposit may be eroded material. Thin layer models of the kind used here have been extended to incorporate mass changed due to erosion.[34] Such a model must address how the erosion of a bed, in turn, changes the local topography, and how such elevation changes are fed back into the numerical solver.

Finally, the Mohr–Coulomb model used here accounts only for the deformation of solids material. Missing is the effect of interstitial fluid—either air, in dilute flows, or water, in muddy flows. Pore fluid may also be important in initiating mass flows. Iverson[7,8] has taken a first step by including fluid into a thin layer model, but more work is required before we can be confident of any approach.

ACKNOWLEDGMENTS

Computational results reported here were performed at the Center for Computational Research at the University at Buffalo. This project is supported in part by Grants No. NSF 0121254 and No. NSF 0087665 from the National Science Foundation.

[1] R. I. Tiling, *Volcanic Hazards*, American Geophysical Union Short Course, 1989, Vol. I.
[2] National Research Council, "The eruption of Nevado del Ruiz Volcano, Colombia, South America, November 13, 1985," The Committee on Natural Disasters, 1991.
[3] National Research Council, "Mount Ranier: Active cascade volcano," The U.S. Geodynamics Committee, 1994.
[4] J. W. Vallance, "Post-glacial lahars and potential hazards in the White Salmon River system on the southwest flank of Mt. Adams, Washington," USGS Bulletin 2161, 1999.
[5] S. B. Savage and K. Hutter, "The motion of a finite mass of granular material down a rough incline," J. Fluid Mech. **199**, 177 (1989).
[6] K. Hutter, M. Siegel, S. B. Savage, and Y. Nohguchi, "Two dimensional spreading of a granular avalanche down an inclined plane; Part 1: Theory," Acta Mech. **100**, 37 (1993).
[7] R. M. Iverson, "The physics of debris flows," Rev. Geophys. **35**, 245 (1997).
[8] R. M. Iverson and R. P. Denlinger, "Flow of variably fluidized granular material across three-dimensional terrain 1. Coulomb mixture theory," J. Geophys. Res. **106**, 537 (2001).
[9] R. P. Denlinger and R. M. Iverson, "Flow of variably fluidized granular material across three-dimensional terrain 2. Numerical predictions and experimental tests," J. Geophys. Res. **106**, 553 (2001).
[10] J. M. N. T. Gray, "Granular flow in partially filled slowly rotating drums," J. Fluid Mech. **441**, 1 (2001).
[11] J. M. N. T. Gray, M. Wieland, and K. Hutter, "Gravity-driven free surface flow of granular avalanches over complex basal terrain," Proc. R. Soc. London, Ser. A **455**, 1841 (1999).

[12]Y.-C. Tai, S. Noelle, and J. M. N. T. Gray, ''Shock-capturing and front tracking methods for granular avalanches,'' J. Comput. Phys. **175**, 269 (2002).
[13]D. G. Schaeffer, ''Instability in the evolution equations describing granular flow,'' J. Diff. Eqns. **66**, 19 (1987).
[14]E. B. Pitman and D. G. Schaeffer, ''Stability of time dependent compressible granular flow in two dimensions,'' Commun. Pure Appl. Math. **40**, 421 (1987).
[15]D. G. Schaeffer and E. B. Pitman, ''Illposedness in three dimensional plastic flow,'' Commun. Pure Appl. Math. **41**, 879 (1988).
[16]K. Terzaghi, ''The shearing resistance of saturated soils and the angle between planes of shear,'' Proceedings of the 1st International Conference on Soil Mechanics, 1936, pp. 54–56.
[17]W. J. M. Rankine, ''On the stability of loose earth,'' Philos. Trans. R. Soc. London **147**, 9 (1857).
[18]H. A. Janssen, ''Versuche uber Getreidedruck in Silozellen,'' Z. Ver. Dtsch. Ing. **39**, 1045 (1895).
[19]R. Nedderman, *Statics and Kinematics of Granular Materials* (Cambridge University Press, Cambridge, 1992).
[20]E. B. Pitman, ''Forces on bins: The effect of random friction,'' Phys. Rev. E **57**, 3170 (1998).
[21]http://www.esri.com
[22]S. F. Davis, ''Simplified second-order Godunov-type methods,'' SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **9**, 445 (1988).
[23]A. Harten, P. D. Lax, and B. van Leer, ''On upstream differencing and Godunov-type schemes for hyperbolic conservation laws,'' SIAM Rev. **25**, 35 (1983).
[24]V. V. Rusanov, ''Calculation of interaction of non-steady shock waves with obstacles,'' J. Comput. Math. **1**, 267 (1961).
[25]R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2002).
[26]E. F. Toro, *Shock Capturing Methods for Free Surface Shallow Flows* (Wiley, New York, 2001).
[27]M. Berger and P. Colella, ''Local adaptive mesh refinement for shock hydrodynamics,'' J. Comput. Phys. **82**, 64 (1989).
[28]J. E. Flaherty, R. M. Loy, M. S. Shephard, M. L. Simone, B. K. Szymanksi, J. D. Teresco, and L. H. Ziantz, ''Distributed octree data structures and local refinement method for the parallel solution of three-dimensional conservation laws,'' in *Grid Generation and Adaptive Algorithms* (Springer-Verlag, Berlin, 1999), IMA Vol. 113.
[29]H. E. Carter and J. F. Stewart, ''The SIERRA Computational mechanics framework,'' SAND2001-2720P, Sandia National Laboratories, Albuquerque, NM, 2001.
[30]A. Laszloffy, J. Long, and A. Patra, ''Simple data management schemes and scheduling schemes for managing the irregularities in parallel adaptive *hp* finite element simulations,'' Parallel Comput. **26**, 1765 (2000).
[31]A. K. Patra, A. C. Bauer, C. C. Nichita, E. B. Pitman, M. F. Sheridan, M. Bursik, B. Rupp, A. Webber, A. Stinton, L. Namikawa, and C. Renschler, ''Parallel adaptive numerical simulation of dry avalanches over natural terrain,'' J. Volcanic Geophys. Res. (to be published).
[32]M. F. Sheridan, A. J. Stinton, A. K. Patra, E. B. Pitman, A. C. Bauer, and C. C. Nichita, ''Evaluating TITAN2D mass-flow model using the 1963 Little Tahoma Peak avalanches, Mount Rainier, Washington,'' J. Volcanol. Geotherm. Res. (to be published).
[33]S. Rodriguez-Elizarraras, C. Siebe, J.-C. Komorowski, J. M. Espindola, and R. Saucedo, ''Field observations of pristine block-and-ash flow deposits emplaced 16–17 April 1991 at Volcan de Colima, Mexico,'' J. Volcanol. Geotherm. Res. **48**, 399 (1991).
[34]E. B. Pitman, C. C. Nichita, A. K. Patra, A. C. Bauer, M. Bursik, and A. Webber, ''A model of granular flows over an erodible surface,'' Discrete Continuous Dyn. Syst., Ser. B **3**, 589 (2003).