Landslide Dynamics In The Philippines:
A Mathematical And Computational Study Using Cellular
Automata And The Minimization Algorithm

Timothy James L. Yusun
tjyusun@yahoo.com
Ateneo de Manila University
Katipunan Avenue, Loyola Heights
Quezon City 1108, Philippines
+63 2 426 6001

Rafael P. Saldaña
rpsaldana@yahoo.com

ABSTRACT
In this paper, we present some mathematical and computational aspects of modeling landslide dynamics in the Philippines. In particular, we discuss cellular automata and the minimization algorithm with implementations using Mathematica. We also present some preliminary computer simulation results that conform with the expected behavior of natural debris flow.

Keywords
Landslide dynamics, cellular automata, minimization algorithm.

1. INTRODUCTION
1.1 Introduction
The Philippines experiences several natural disasters every year. Among these calamities are landslides and debris flows.

At 10:00am on February 17, 2006, the village of Guinsaugon in the province of Southern Leyte was hit by a landslide. At 10:00pm later that day, another landslide hit Guinsaugon, prompting nearby villages to evacuate as a preventive measure. Houses and schools were buried in the mud and debris along with hundreds of people, causing countries and organizations from all over the world to assist in the relief and rehabilitation efforts [10].

In a country such as the Philippines, where landslides carry disastrous effects on the economy, on infrastructure, and on the lives of citizens, it is imperative that the dynamics behind these geological hazards are understood.

For landslides which have already happened, various books discussing landslide analysis offer a methodology of slope stability analysis: this includes data gathering and an on-site investigation of the affected area, a geological mapping of sliding slopes, and the measurement of relevant quantities like stress, shearing forces, and the factor of safety. After this, corrective and preventive measures are implemented based on the findings [1 & 13]. One weakness of this evaluation process is that there are too many points to focus on, i.e. the phenomenon is too complex a system. Aside from the qualitative geological observations on the area involved, numerous other elements are to be computed, including the aforementioned factor of safety and residual horizontal stress. As a result, much information has to be processed before making any general conclusions on the landslide dynamics. Moreover, though a solution to a system of differential equations which govern the landslide behavior can ideally be numerically approximated (as finding analytical solutions will be nearly impossible for all but a few, simple cases), it will not be an easy method to implement. As such, some authors have since been proposing the use of Cellular Automata (CA) models for flow type landslides [3].
Fig. 2 SAR-PALSAR northwest view of Leyte Island showing the affected area [5]

1.2 Statement of the Problem

This study aims to:

1. elucidate on the minimization algorithm as applied to landslide dynamics;
2. develop computer simulations of landslide dynamics using a cellular automata model based on the work of D’Ambrosio and co-workers, and the Mathematica programming language.

1.3 Organization of Study

This paper will be organized as follows: Chapter 2 talks about the dynamics of landslides, Chapter 3 gives an introduction to cellular automata, Chapter 4 will be a more thorough discussion of the minimization algorithm used in the model, Chapter 5 will incorporate this algorithm with other parameters in order to formulate a model, Chapter 6 outlines some results and discussions, while Chapter 7 concludes the paper.

2. LANDSLIDE DYNAMICS

2.1 Basic Definitions

Landslides are defined as “movements of sliding rocks, separated from the underlying stationary part of the slope by a definite plane of separation” (p.1). Analyses of these hazards fall into two categories: the geologist studies the causes of landslides, factors affecting movements, and the resulting surface forms, while the engineer attempts to measure the safety of the slope with respect to construction projects or existing structures already situated there [13]. In actuality, both approaches depend on each other to produce a thorough study.

The primary factor which causes landslides is the gradient or steepness of the slope [1]. Recall that the magnitude of force in a direction is simply the product of the mass of the material and its acceleration. For a downslope, the acceleration due to gravity (due to the weight of the material) can be separated into a parallel and a perpendicular component.

Thus, as the angle of elevation \( \theta \) increases, the perpendicular component increases with it, and the risk of slope failure. Alternatively, an increase in the mass of the material may cause failure even if \( \theta \) is constant. Deposition of soil, precipitation, and an increase in the amount of groundwater may all lead to an increase in mass (or mass density), which if sufficient to overcome the shearing forces at work in the slope, may cause a slide or a flow [1].

Fig. 3 Parallel and perpendicular components of force

2.2 Classifications and Types of Landslides

Generally, while slope gradient does play a huge role in landslide dynamics, there are numerous other factors which determine the behavior of a failure in slope, or whether there will be any failure at all. Considerations such as the material involved and the surrounding climate will also be influential. To make their analyses easier, geologists resort to grouping landslides based on these factors and more; here are some classifications of landslides [1].

- Material – Landslides may be composed of rock material, soil, ice, or a mixture of these. Correspondingly, the types of rock (e.g. granite, sandstone), the types of soil (e.g. clay, loam, gravel), and the composition of these elements are also determined.

- Velocity – Economically, determining the speed of a landslide gives a measure of how long people will have to evacuate, and how extensive the structural damage will be. Some landslides occur at an extremely slow rate, taking months at a time to move a few meters, but there are some which take place suddenly, without any previous warning. In general, landslides fall into three categories: rapid (seconds to minutes), intermediate (minutes to hours), and slow (days to years).

- Displacement – The distance a landslide travels may vary from meters to kilometers, depending on the material involved, its velocity, precipitation, and other factors.

- Mechanism – Landslides can also be characterized by the interaction which takes place between the sliding mass of material and the stationary part of the slope underneath. When the sliding mass can be clearly distinguished from the stationary part, for instance loose soil moving over bedrock, the landslide is called a slide. When the two components are such that they almost cannot be distinguished from one another due to the slow, almost viscous liquid-like flow of the mass, the landslide is a flow. Conversely, when the sliding mass all but loses coherent contact with the slope, as rocks do when they roll down a hill, the landslide is a rockfall.

Although there are many ways of characterizing landslides, in general all of these factors contribute to each other’s effects.
2.3 Landslide Factors

As mentioned, the gradient of the slope under consideration plays a huge role in causing landslides. However, there are numerous other factors which may or may not directly affect the slope gradient, but nevertheless affect the behavior of the landslide. Aside from the addition of mass, some factors include [13]:

- **Water content** – Precipitation or other sudden changes in water content may destabilize slopes because water increases the pressure on the soil and decreases internal friction and cohesion. Also, such a change may effect a displacement of particles.

- **Ground water** – A greater amount of ground water may also lead to an increase in hydrostatic pressure and thus instability.

- **Shocks and vibrations** – Earthquakes are perennial causes of landslides, since they disrupt the equilibrium conditions of the slope, and may decrease friction/loosen up some parts. Some studies have actually measured the correlation between earthquake, typhoon, and landslide occurrences [6]

- **Vegetation cover** – Deforestation is another cause of landslides, since the roots of trees lend stability to the slope.

Taking these into account, a model may be formulated that will include these factors, and other mechanics considerations discussed earlier. These may eventually be simplified using other physics or fluid dynamics principles, to make simulations easier.

3. CELLULAR AUTOMATA

3.1 A General Overview

Taken generally, a CA model necessarily includes three components: cells, states, and transitions. The cells (which together form a cellular space) are homogeneous, and evolve according to their [initial] states and a transition function, which defines the behavior of the system. The whole space changes in discrete time steps, the data corresponding to each cell constantly being updated.

This approach involves taking the area under consideration and dividing it into cells. Usually, these cells take on a uniform shape and size, which are determined by the physical characteristics of the phenomena being modeled. This results in a lattice, the specific arrangement of cells in the grid. Some examples are the usual m x n matrix, or the hexagonal lattice used in [2].

Fig. 4 & 5 A hexagonal lattice and a square lattice

To compute the state of a cell (say its height) for the next time step, the transition rule often takes as its input the state (heights) of the cells in a neighborhood of that cell, i.e. those immediately surrounding it [9], then computing using an algorithm of some sort the new state (height) of the central cell under consideration. The hexagonal lattice above would employ a neighborhood with 7 cells (one central cell and the six around it), while a square lattice might use a Von Neumann neighborhood or a Moore neighborhood [11].

Fig. 6 & 7 A Von Nuemann and a Moore neighborhood

One selling point of the CA-based approach to modelling is its characteristic of acentrism, or that the global evolution of the system may be described in terms of local interactions [2]. Interestingly, Stephen Wolfram in his book A New Kind of Science explains in detail the many applications of automaton – fractals, biology, fluid flow, even finance. Here he describes his wonder at how very simple rules can produce very complex results, and that with the arrival of computers on the scientific scene, many complex systems can actually be modelled and studied without much difficulty, seeing how simple their underlying rules seem to be [11]. The ease with which computations and simulations can be done stems from the acentric property of CA.

Fluid dynamics seems to be a prime candidate for the cellular automata-based approach, seeing how the interactions between particles in a system may be described discretely [11]. A viable alternative to analytical methods, it should be noted nonetheless that taking the continuum limit of the model will result once again in the realm of difference equations [2].
3.2 A Formal Definition

Formally, a cellular automaton is a 4-tuplet \((L, S, N, s)\), where \([9]\)

- \(L = \{c(i,j)|m,n \in \cdot, 0 \leq i \leq m, 0 \leq j \leq n\}\) is an \(m \times n\) lattice with \(c(i,j)\) being the cell in the \(i\)th row and \(j\)th column;
- \(S = \{s_1, s_2, ..., s_r\}\) is the set of possible states of a cell in the lattice;
- \(N(i,j)\) is the neighborhood of cell \(c(i,j)\) at time \(t\) (which includes the states of the cell and its neighbors);
- \(\sigma(N(i,j))\) is the transition function which gives the new state of cell \(c(i,j)\) at time \(t + 1\).

The choices for the parameters \(S\) and the transition rule \(s\) is determined by the physical phenomenon being studied; for landslides, \(S\) may include factors such as thickness of debris, water content and temperature, while \(s\) may compute for the new values of these states depending on the physics of the system.

4. THE MINIMIZATION ALGORITHM

4.1 Introduction

In landslide modeling, the CA model is a two-dimensional lattice of the area under consideration, where the height of each cell is considered to be a substate to reduce the number of dimensions. In [2], other substates were included, e.g. the thickness of debris, energy of debris, and depth of erodable soil cover. Particularly, the computation of debris outflows into adjacent cells will be of huge relevance. In the same paper, D’ Ambrosio et al. describe a strategy based on the hydrostatic equilibrium principle to compute precisely this.

For simplicity, first take a single CA cell in the lattice, which we will call the “central cell.” Now the debris outflows coming from this central cell will be limited to those cells in its immediate neighborhood. Say there are \(m - 1\) neighboring cells, and denote them by the indexes \(1, 2, ..., m - 1\). The inflows these neighboring cells will receive depend on the difference in heights (for instance, debris will flow from a cell with debris thickness 10 to one with debris thickness 3).

At time step \(t\), two quantities of the central cell are determined: the mobile part \(q[0]\) and the fixed part \(q[1]\) of its total height. Here \(p\) represents the total “amount” of material in the central cell which can move to neighboring cells and \(q[0]\) is the fixed part – debris thickness and the altitude due to bedrock in the case of landslides, respectively.

The total height of the central cell is then \(p + q[0]\). Now let \(q[i]\) denote the height of the \(i\)th neighboring cell. Note that differentiating between the mobile part and the fixed part of neighboring cells is not a concern, since they only receive inflows from the central cell for this simple case.

At time step \(t + 1\), denote the flows from the central cell to the \(i\)th neighboring cell by \(f[i] (0 \leq i \leq m - 1)\); \(f[0]\) is the part of \(p\) that is not distributed, and their corresponding new heights by \(q'[i] = f[i] + q[i]\), for \(0 = i = m - 1\). Furthermore, denote by \(q'[\min]\) the minimum value of the \(q'[i]\). The \(f[i]’s\) are determined by minimizing the following expression:

\[
\sum_{i=0}^{m-1} (q'[i] - q''[\min])
\]

To see why minimizing this expression is equivalent to determining the outflows, note that this summation represents the total amount of variation present in the new configuration, its “variance,” so to speak. A higher variance is equivalent to larger differences of heights between cells, and thus a higher pressure difference. Since the system will naturally tend to equilibrium, i.e. the configuration with the least amount of pressure differences between cells, computing the outflows from the central cell is the same as minimizing the expression above. For instance, consider the simple case of the following initial configuration \((t = 0; \text{assume } q[0] = 0)\).

\[
\begin{array}{ccc}
7 & 6 & 2 \\
\end{array}
\]

Fig. 8 A simple configuration, \(t = 0\)

Again, recall that only outflows coming from the central cell will be considered. Since \(q[1] > p\), any flow from cell 0 to cell 1 will result in increasing the value of the summation described above, particularly \(q'[1] - q'[\min]\), since cell 1 is not the cell with the minimum height. Therefore it can be deduced that in the next time step, there will be no outflow from the central cell to cell 1, that is, \(f[1] = 0\). This line of reasoning is not unlike the reason why water does not flow upward, but downward – fluids tend to behave as if to minimize pressure differences, or tend towards equilibrium. For \(f[2]\), notice that the expression \(q'[2] - q'[\min]\) is minimized when \(f[2] = 2\), or when \(q'[0] = q'[2] = 4\). The configuration at \(t = 1\) is therefore:

\[
\begin{array}{ccc}
7 & 4 & 4 \\
q'[1] & q'[0] & q'[2]
\end{array}
\]
Fig. 9 A simple configuration, $t = 1$

For a neighborhood with more than 2 cells, a similar procedure is employed. First, cells whose heights are greater than that of the central cell do not receive inflow. For the remaining cells (say there are $k$ cells whose heights are less than $p + q[0]$), the corresponding $f[i]$’s are computed, with some possibly 0, such that the differences are minimized.

Observe that the summation (1) becomes equal to 0 in the case where $q'[i] = q^r \min$ for all $i$, or equivalently, $q'[0] = q^r[1] = \ldots = q^r[m-1]$. This can occur in the case when $q[0]$ is very large compared to the $q[i]$’s. Here, the final configuration is a flat neighborhood of $m$ cells, whose heights are the average of all the initial heights. This insight about the average height offers a way to minimize the differences in the general case: for the $k$ remaining cells, compute their average height, and call this value $H$. If there is any noncentral cell, say cell $j$ for some $j \leq 0$, whose height $q[j]$ is greater than $H$, then setting all cell heights equal to $H$ will imply an outflow from cell $j$, which is not allowed. Similarly, if the fixed part of the central cell, $q[0]$, is greater than $H$, then setting all cell heights equal to $H$ will imply that $H = q'[0] < q[0]$, or that some portion of the fixed part of the central cell was transferred to its neighbors, which is again impossible. From these two cases, all cells $i$ such that $q[i] > H$ will not receive any inflow. Now the new value of $H$ is computed for the remaining cells, and the steps above are repeated until all the remaining cells are assured of receiving some inflow. The heights of these cells are set to be equal to the most recent value of $H$, and thus the new configuration is achieved. This algorithm is summarized below, using the algorithm stated in [2] with some modifications.

Minimization Algorithm for Computing Debris Outflows

0. Let $A$ is the set of all cells which may receive flows.

Initially, $A$ contains cells 0, 1, ..., $m - 1$.

1. Cells $i$ with $q[i] = p + q[0]$ are eliminated from $A$.

2. The average $H$ is computed, considering only cells in $A$.

3. Cells $i$ with $q[i] = H$ are eliminated from $A$. Go back to step 2 if any cell was eliminated.

4. The flows $f[i]$ for $0 \leq i \leq m - 1$ are computed as follows:

$$f[i] = \begin{cases} H - q[i] & \text{if } i \in A \\ 0 & \text{if } i \notin A \end{cases}$$

4.2 Proof of the Algorithm

Does this algorithm minimize (1) in all possible cases? The following theorem was outlined in [4], the proof of which is made easier by the use of Lemma 1 below.

**Theorem 1**

The minimization algorithm described above minimizes the expression in (1).

**Lemma 1**

If $q'[i]$ (for $i = 0, 1, 2, \ldots, m - 1$) denotes the configuration obtained by using the minimization algorithm, and $r'[i]$ denotes any other different configuration (for $i = 0, 1, 2, \ldots, m - 1$), then $r'_\min < q'_\min$.

**Proof of Lemma 1.**

Note that the statement $p = \sum_{i=0}^{m-1} f[i]$ holds for any possible configuration since $p$ is the mobile part of the central cell, and is the part that is distributed among its neighbors. Thus the sum of flows is just equal to $p$, that is, always constant.

Let $f[i]$ denote the flows associated with $q'[i]$, and $g[i]$ the flows associated with $r'[i]$. Then,

$$\sum_{i=0}^{m-1} f[i] = \sum_{i=0}^{m-1} g[i].$$

Since the two configurations $q'[i]$ and $r'[i]$ are different, there must exist some $j, k$ in the set $\{1, 2, 3, \ldots, m - 1\}$ such that $f[j] > g[j]$ and $f[k] < g[k]$. That is, decreasing some flows will increase the outflow to other cells in the neighborhood – the conservation of mass principle. In effect, what this is saying is that there will be some $j$ such that the outflow to cell $j$ is decreased in the configuration $r'[i]$.

Note the following statements:

(a) $q'_\min = H$

(b) If $i \in A$, then $q'[i] = H$ and $f[i] = H - q[i]$.

(c) If $i \notin A$, then $q'[i] = q[i]$ and $f[i] = 0$.

Statement (a) is true because all cells which did not receive flows already had a higher height, and cells which did receive flows (whose heights were less than $H$) obtained a new height of $H$. Both (b) and (c) are results of the algorithm.

Now, since $f[j] > g[j] \geq 0$, the cell $j$ must have originally received some flow, and $j \in A$, then $q'[j] = H = q'_\min$. Since $f[j] > g[j]$, adding $q'[j]$ to both sides of the inequality yields $q'[j] + f[j] > q[j] + g[j]$, and thus $q'_\min = q'[j] > r'[j]$. Thus there must be at least one cell in the new configuration $r'[i]$ such that its height is less than $q'_\min$, and so $r'_\min < q'_\min$.

The proof of Theorem 1 immediately follows:

**Proof of Theorem 1**

Let $q'[i]$ denote the configuration obtained using the algorithm previously described, and $r'[i]$ any other configuration ($i = 0, 1, 2, \ldots, m - 1$). To prove that the algorithm really minimizes (1), it is sufficient to show the following identity:

$$\sum_{i=0}^{m-1} (r'[i] - r'_\min) \geq \sum_{i=0}^{m-1} (q'[i] - q'_\min)$$

(2)

This can be shown by manipulating both sides of (2):
Example 1

$$\sum_{i=0}^{m-1} (r'[i] - r'_{\min}) = \sum_{i=0}^{m-1} (g[i] + g'[i]) - \sum_{i=0}^{m-1} r'_{\min} = \sum_{i=0}^{m-1} g[i] + p - m (r'_{\min})$$

$$\sum_{i=0}^{m-1} (q'[i] - q'_{\min}) = \sum_{i=0}^{m-1} (q[i] + f[i]) - \sum_{i=0}^{m-1} q'_{\min} = \sum_{i=0}^{m-1} q[i] + p - m (q'_{\min})$$

At t = 0:

$$q_{th1} = \begin{bmatrix} 2 & 6 & 5 \\ 24 & 26 & 0 \\ 9 & 12 & 10 \end{bmatrix}, \quad qa1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 3 & 1 & 1 \end{bmatrix}$$

At t = 1:

$$fm = \begin{bmatrix} 9 & 9 & 9 \\ 24 & 9 & 9 \\ 12 & 13 & 11 \end{bmatrix}, \quad \text{fm} = \begin{bmatrix} 6 & 2 & 3 \\ 0 & 7 & 8 \\ 0 & 0 & 0 \end{bmatrix}$$

The second-to-the-last line follows from Lemma 1.

4.3 Mathematica Implementation

Mathematica is a computing system that “provides cross-platform support for tasks such as symbolic or numerical calculations, arbitrary precision arithmetic, data processing, and plotting.” It also has a programming language which “supports functional and procedural programming styles” [7]. Released in 1988, the breakthrough program “marked the beginning of modern technical computing,” aiming to create one unified system that could work with any aspect of computing. At present, the depth and breadth of Mathematica can be seen in the way it is applied to not only mathematical and technical computing, but to other fields as well [12].

The minimization algorithm was implemented in Mathematica by considering a 3 x 3 matrix of values. A Moore neighborhood was used, in which the 8 cells which surround the central cell are all considered adjacent to cell 0.

The two inputs are qth1 and qa1, the mobile parts and the fixed parts of each cell, respectively. The program outputs the flow matrix fm and the new configuration for the neighborhood (total heights of the cells). The simple program was run for sample configurations, as shown below:

Example 1

At t = 0: $q_{th1} = \begin{bmatrix} 2 & 5 & 5 \\ 4 & 50 & 3 \\ 2 & 1 & 0 \end{bmatrix}, \quad qa1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

At t = 1: $fm = \begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix}, \quad \text{fm} = \begin{bmatrix} 6 & 3 & 3 \\ 4 & 8 & 5 \\ 6 & 7 & 8 \end{bmatrix}$

This first example shows that averaging minimizes the expression in (1), for those cells which can receive inflows.

Example 2

Here example 2 is repeated but with the altitude of the central cell changed to 40. Note that the new configuration has 40 as the central cell’s height – the whole of its mobile part has been distributed to the adjacent cells, and only the fixed part is left.

The program above applies the algorithm and computes debris flows only for 1 neighborhood. For a larger configuration with more cells, the algorithm is applied to all possible 3 x 3 neighborhoods taken one at a time, and the new configuration is given by the sum of all matrices fm for corresponding cells. For instance, in the example below, a 3 x 4 matrix undergoes the program (assume for simplicity that the altitude of all the cells is 0).

Example 3

At t = 0: $q_{th1} = \begin{bmatrix} 2 & 6 & 5 \\ 24 & 26 & 0 \\ 9 & 12 & 10 \end{bmatrix}, \quad qa1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 40 & 1 \\ 3 & 1 & 1 \end{bmatrix}$

At t = 1: $fm = \begin{bmatrix} 43/4 & 43/4 & 43/4 \\ 24 & 40 & 43/4 \\ 12 & 13 & 11 \end{bmatrix}, \quad \text{fm} = \begin{bmatrix} 31/4 & 15/4 & 19/4 \\ 0 & 0 & 39/4 \\ 0 & 0 & 0 \end{bmatrix}$

Example 4

At t = 0: $q_{th1} = \begin{bmatrix} 1 & 3 & 2 & 1 \\ 5 & 16 & 10 & 0 \end{bmatrix}, \quad qa1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

At t = 1: $fm = \begin{bmatrix} 30/7 & 30/7 & 67/14 & 5/2 \\ 30/7 & 30/7 & 81/14 & 5/2 \\ 5/2 & 5/2 & 5/2 & 5/2 \\ 2/7 & 9/7 & 67/14 & 3/2 \end{bmatrix}$
5. MODEL FORMULATION

5.1 A CA Model for Debris Flow

A simple CA model based on the minimization algorithm above is defined as the 4-tuplet \((L, Q, N, s)\), where:
- \(L = \{c(i, j)\} \text{ for } 0 < i \leq m, 0 < j \leq n\) is an \(m \times n\) lattice with \(c(i, j)\) being the cell in the \(i\)th row and \(j\)th column;
- \(Q = \{q_{da}, q_a\}\) is the set of possible states of a cell in the lattice, where:
  - \(q_{da}\) is the debris thickness, and
  - \(q_a\) is the altitude;
- \(N(i, j) = \{c(k, l)\} : \|k - i\| \leq 1 \& \|l - j\| \leq 1\) is the neighborhood of cell \(c(i, j)\), a Moore neighborhood with 9 cells; and
- \(\sigma(N(i, j))Q^k \rightarrow Q\) is the transition rule for the set of states of the cells of \(L\), or more specifically, a neighborhood of cells.

The initial values of \(q_{da}\) and \(q_a\) are entered in input matrices, all possible neighborhoods are considered, and the minimization algorithm is applied to each of these to obtain the cumulative flow matrix of values \(F(i, j)\). This matrix is then designated as the new debris thickness.

5.2 A Second CA Model for Debris Flow

A more accurate model can be obtained by including other factors which may affect the behavior of a landslide. One is described by D’Ambrosio, et. al., in [2], and which is outlined below with some modifications (e.g. the paper used a hexagonal lattice):

\[ \text{SCIDICA S3-hex} = < L, X, Q, P, e, g > \text{, where:} \]
- \(L = \{c(i, j)\} \text{ for } 0 < i \leq m, 0 < j \leq n\) is an \(m \times n\) lattice with \(c(i, j)\) being the cell in the \(i\)th row and \(j\)th column;
- \(N(i, j) = \{c(k, l)\} : \|k - i\| \leq 1 \& \|l - j\| \leq 1\) is the neighborhood of cell \(c(i, j)\), a Moore neighborhood with 9 cells.
- \(Q = Q_a \times Q_{da} \times Q_s \times Q_o \times Q_m\) is a set of states of each cell, with
  - \(q_a\) is the cell altitude,
  - \(q_{da}\) is the debris thickness,
  - \(q_s\) is the energy of the debris,
  - \(q_{da}\) is the depth of erodable soil cover, and
  - \(q_o\) are the debris outflows.
- \(P = \{p, p, p_{ad}, p_{af}, p_{a}, p_{af}, p_{am}, p_{ar}\}\) is the set of global parameters defined for the model, as follows:
  - \(p_{a}\) is the side length of a cell,
  - \(p_{ad}\) is the time correspondence of a CA step,
  - \(p_{af}\) is the amount of debris which is unmovable (adhesion effects),
  - \(p_{af}\) is the height threshold for outflows,
  - \(p_{af}\) is the relaxation rate for outflows,
  - \(p_{am}\) is the run-up loss due to friction,
  - \(p_{ar}\) is the activation threshold for mobilisation of soil cover, and
  - \(p_{ar}\) is the parameter of progressive erosion of the soil cover.

\(\sigma : Q^k \rightarrow Q\) is the transition function, which includes the following steps:
1. mobilisation triggering and effects,
2. debris outflows,
3. update of debris thickness and energy, and
4. energy loss.

This model is also more precise, as all the global parameters involved are stated, for instance cell size and time step correspondence (which are implicitly assumed in the previous model). The main addition to this model is the run-up and energy factors.

A cellular automata-based model is a discrete description of a natural, continuous phenomenon. D’Ambrosio, et. al., derive a method for describing velocity and energy in CA, based on a paper by Marchi and Rubatta (1981). The former use fluid dynamics considerations (where the kinetic energy of a fluid is given by \(v^2/2g\)) to develop a similar quantity for a CA context, the run-up, or the “height that can be reached by the flow” [2].

If the height of a cell is given by \(h\), the run-up is defined as \(h + \Delta h\), where \(\Delta h\) is the “kinetic head,” and is equal to \(v^2/2g\), where \(v\) is the speed of the flow, and \(g\) is the constant acceleration due to gravity. Now assume that a cell has mass \(m\), height \(h\), base with area \(A\), and that the material it is composed of has density \(\rho\). Also, the potential energy of the cell is given by:

\[ U = \rho g A \int_{z=0}^{h} \frac{g A}{2} \, \frac{d\varepsilon}{dz} = \frac{\rho g A}{2} h^2 \]

To incorporate the effect of the kinetic head, the height is virtually increased to \(r = h + \Delta h\), in the process (since mass is conserved) decreasing the density of the cell to some \(\varepsilon = h/r < \rho\). Thus the energy increase is

\[ U' = \frac{\rho' g A}{2} r^2 = \left(\frac{h}{r}\right) \frac{g A}{2} r^2 = \frac{\rho g A}{2} h + \frac{h^2}{2g} > U \]

Thus, for the CA model, \(q_{o} = \left(\frac{\rho g A}{2} \right) q_{da} \cdot r = k \cdot q_{da} \cdot r\) for a constant \(k\). Moreover, this equation shows that \(r\) is proportional to \(q_{da}\).

For the transition rule which computes debris outflows (#2 above), the height of the central cell is virtually increased to \(r\), to
consider run-up effects. Using the minimization algorithm in Chapter 4, the following are the values of the variables: \( q[0] = q_{th}(0) + p_{w} \), \( p = r - p_{w} \) (adhesion is factored in), and \( q[i] = q_{th}(i) + q_{ad}(i) \) for \( i = 1, 2, 3, \ldots, 8 \).

Because of the virtual increase, outflows have to be normalized (brought back to the correct scale) by a factor \( v_{ad} = h / r = k q_{th}(0) / q_{th}(i) \), so that the outflow from the central cell to cell \( i \) is just equal to \( v_{ad} f[i] \).

For transition rule #3, the new debris thickness \( q_{th}^{'} \) can be obtained by summing up all inflows to a cell and subtracting all outflows, adding this to the original thickness \( q_{th} \). For the energy \( q_{e} \), energy variations due to the flows are considered:

\[
q_{e} = \left( q_{th} - \text{sum of all outflows} \right) \left( \frac{q(0)}{q_{th}(0)} \right) + \sum_{i=1}^{6} \text{inflow from cell } i \left( \frac{q(i)}{q_{th}(i)} \right)
\]

For transition rule #1, the condition that will trigger erosion is \( q_{e}(0) > p_{w} \), so that if \( (q_{e}(0) - p_{w}) p_{w} < q_{th}(0) \), then the quantity of eroded soil \( d \) is the left-hand side of this inequality, while if \( (q_{e}(0) - p_{w}) p_{w} > q_{th}(0) \), then \( d = q_{th}(0) \). Thus the following are updated:

- Cell altitude: \( q_{th}^{'} = q_{th}(0) - \Delta_{d} \)
- Depth of erodible soil cover: \( q_{d}^{'} = q_{d}(0) - \Delta_{d} \)
- Debris thickness: \( q_{th}^{'} = q_{th}(0) - \Delta_{d} \)
- Run-up: \( r^{'} = r + \Delta_{d} \)
- Energy: \( q_{e}^{'} = k \left( \frac{q_{th}}{q_{th}(0)} \right) (r) = k \left( q_{th}(0) + \Delta_{d} \right) (r + \Delta_{d}) \)

The last transition rule takes into account energy loss by friction, and computes this using the parameter \( p_{r} \) as follows:

\[
\Delta r = \begin{cases} 
  p_{r}, & \text{if } k \frac{q_{th}(0)}{q_{th}(0)} - p_{r} > q_{th}(0) \\
  k \frac{q_{th}(0)}{q_{th}(0)} - q_{th}(0), & \text{otherwise}
\end{cases}
\]

In light of this change in run-up, the energy of the cell is updated to \( q_{e}^{'} = q_{e}(0) - k \cdot r q_{th}(0) \).

6. RESULTS AND DISCUSSIONS

A program was written to implement the first CA model described above, which utilizes the minimization algorithm and applies it to an lattice. Input values were \( q_{th}, q_{a}, \) the matrix of cell debris thickness, \( q_{a}, \) the matrix of cell altitude, and \( t_{m}, \) the number of time steps the algorithm would be applied. The function Landslide[\( q_{th}, q_{a}, t_{m} \)] outputs \( t_{m} + 1 \) raster plots of the cell heights in the lattice (the corresponding debris thickness plus the altitude of each cell). The function could also be modified to output 3D List Plots, accordingly. The code for the whole Mathematica 5.0 program can be found in Appendix A.

Below are some preliminary results on simulations done using Mathematica.

Example 1: The Slope

The first example involves a simple slope (from the left to the right), and debris on top of that slope. As expected, after some time, the debris was already at the bottom of the slope. The program was run for \( t = 20 \) time steps, and \( q_{th} \) and \( q_{a} \) were constructed in Microsoft Excel. Figure 10 shows the raster plots generated by Mathematica, while Fig. 11 shows some 3D plots generated at different times. For the raster plots, a darker cell corresponds to a larger height.

![Fig. 10 Example 1: Raster plots at various times](image)

![Fig. 11 Example 1: 3D plots at various times](image)
low-altitude area. Again, the following figures show a raster plot of the simulation. Note the color of the upper-left area, which changes from white to gray as debris flows in.

![Fig. 12 Example 2: Raster plots at t = 0, 12, 30, 50, 100, 150](image)

**Example 3: The Hill**

For this case, debris was placed on top of a hill. As results show, the flow immediately started to descend from the hill, and spread towards all sides in a fluid-like manner.

![Fig. 13 3D plots of a hill at various times](image)

7. **CONCLUSION AND RECOMMENDATIONS**

In this paper, we discussed the mathematics behind the minimization algorithm. We also approximated landslide dynamics using a cellular automata-based numerical method, the minimization algorithm, and Mathematica. Some computer simulation results that conform with expected natural debris flow were presented.

To come up with more valid results, actual data such as debris thickness, altitude, amount of soil cover, etc. from the Guinsaugon and other Philippine landslides are needed. A sample raster plot which describes elevation data for the country was already obtained, and future developments could incorporate this new information\(^1\). The plot is called a digital elevation model (DEM) and will need a special kind of software before it can be read.

![Fig. 14 DEM of a portion of the Philippines](image)

8. **REFERENCES**


---

\(^1\) Thanks to the **klima Climate Change Center** at the Manila Observatory (Ateneo de Manila University)
http://www.disasterscharter.org/disasters/CALLID_114_e.html


