Cell-DEVS Implementation of the SCIDDICA Landslide Simulation Model

SYSC 5104 Fall 2007
Brian Webb 100727967

Overview

SCIDDICA, the Simulation through Computational Innovative methods for the Detection of Debris flow paths using Interactive Cellular Automata we developed by researchers at the University of Calabria and the Institute of Cybernetics at the Itallian National Research Council. The purpose of SCIDDICA is to provide an accurate model of landslides and avalanches.

In 2003 a version of SCIDDICA called S3-hex was developed and tested[2] against several actual landslides, including the 1998 Campiana and 1999 San Martino slides in Italy. S3-hex includes refinements to the SCIDDICA rules and the definitions of those rules for a hexagonal cell space.

The Cell-DEVS implementation of SCIDDICA uses the S3-hex rules on a square cell space.

It was not possible to completely implement the SCIDDICA rules in Cell-DEVS. The rule for calculating debris outflow from a cell requires itteration, which Cell-DEVS cannot handle. The S3-hex rule is as follows:

B[i] is a function of the central cell (self)'s height and debris thickness and the neighboring cell's height and debris thickness.
pf is the height threashold (related to friction angle) for debris outflow.
rho is a function of the energy and debris in the central cell

1. For all cells in the neighborhood (except self) if B[i] > pf add cell i to set A
2. Calculate average outflow to cells in A:
   rho + sum(height[i] + debris[i]) / cardinality of A
3. If height[i] + debris[i] >= average, remove cell i from A
4. If any cells were removed from A, go back to step 2.
5. for all cells in neighborhood:
   if i is in A: outflow = average - (height[i]+debris[i])
   else: 0

Cell-DEVS has no temporary variables which would allow the creation of the set A inside a rule, and no flow control structures which would allow step 4 to be implemented. Therefore, the rule must be modified to be used with Cell-DEVS. The modified rule is as follows:

For all cells in the neighborhood, if B[i] > pf and (height[i] + debris[i]) < average outflow
    outflow = average - (height[i] + debris[i])
  else
    outflow = 0

Also the SCIDDICA papers do not mention the values for the parameters in their test simulations. The parameter values used here are arbitrary,

Specifications

The SCIDDICA model is a 2D model with N internal state variables, represented in Cell-DEVS as a 3D model with cells on the z axis acting as the state variables. There are also 6 parameters:

- p_time The discrete time step taken by the CA
- p_adhesion The unmovable amount of debris
- pf Height threshold (related to friction angle) for outflows

- p_runup_loss Frictional energy loss
- p_mt Mobilisation threshold for the soil cover
- p_erosion Progressive erosion of the soil cover

The state variables are:

- Altitude: The combinded height of the erodable and non-erodable material in a cell.
- Erodable Soil Depth: The thickness of the erodable material in a cell.
- Energy: The kinetic energy of the debris (landslide material) in the cell.
- Debris Thickness: How deep the debris, the landslide material, is in the cell.
- Outflows: How much debris goes to each cell in the neighborhood (8 of these).
- Kinetic Runup: Used to calculate the energy and outflows.

Some expressions used in the rules

- delta_d: The amount of soil eroded from a cell by the debris. 0 if there is no soil.
- delta_r: The change in runup energy.
  - equals the runup loss parameter (p_rl) if (k* energy/debris - p_rl) > debris
  - else equals (k*energy/debris) - debris
- k: 2 / (rho * g * A) : rho = debris density, g = gravitational acceleration, A = cell area

The rules:

Altitude:

The altitude changes if soil was eroded from the cell.

1. new altitude = old altitude - delta_d

Erodable Soil Depth:

The soil depth changes if there is both debris and energy in the cell, as the moving debris will erode more and more soil over time.

1. new depth = old depth - delta_d

Energy:

The energy changes as the debris picks up more mass from eroding soil, and more momentum. It is also lost due to friction and eroding soil. There are 3 rules for calculating the energy of a cell in SCIDDICA.

Energy lost ot debris outflow, plus energy gained from inflow:

$$E_{new} = (Debris - \sum outflow_i) * (\frac{E}{Debris}) + \sum (inflow_i * (\frac{E_i}{Debris_i}))$$

Potential energy gained from new mass of soil eroded

$$E_{new} = E + E * (\frac{deltad}{Debris}) + k * (debris * deltad + deltad^2)$$

Energy lost to friction

$$E_{new} = E - k * deltar * Debris$$

Debris Thickness:

Debris thickness changes as the debris flows into and out of a cell, and as the soil in the cell is eroded.

1. For all neighboring cells, subtract the outflow to those cells from the inflow from those cells and add the results to the thickness.
2. Add delta_d to the thickness

Kinetic Runup:

The runup is a function of the energy of the cell, the debris thickness and the constant k.

1. runup = k * energy / debris

Debris Outflow to cell i:

The amount of debris that flows from the central cell to cell i in the neighborhood

1. For all cells where ((Altitude[0] + Runup[0]) - (Altitude[i]+Debris[i])) > pf and where the outflow would not be negative:
    1. outflow_to_i = (((k*energy*debris - soil adhesion) + SUM(Altitude[j] + Debris[j]))/ # of cells who meet the criteria) - (Altitude[i]+Debris[i])
2. Otherwise outflow_to_i = 0

Formal Specification

There are 13 planes to represent the state variables:

altitude	plane 0
soil depth	plane 1
energy	plane 2
debris thickness	plane 3
8 debris outflows	planes 4 through 11
runnup	plane 12

The neighborhoods are nearest neighbor neighborhoods for all planes. Technically soil depth and runnup need only themselves and the their corresponding energy and debris thickness but, because Cell-DEVS does not allow for multiple, indpendent neighborhood definitions for each zone, they use the same nearest neighbor neighborhood as all the other state planes.

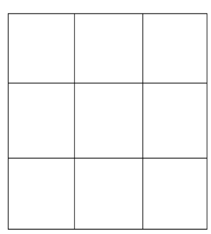

figure 1. Cell Neighborhood, single plane

The time base of the SCIDDICA model is defined by SCIDDICA as an adjustable parameter, which is constant for each simulation of the model. 1 second is used here.

Altitude (plane 0):

$CA = \langle S,n,C,N,T,\tau,qZ_0^+\rangle$

$S = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$.

$n = (1,1,3)$

$C = 0.0 \rightarrow \infty$ (all positive real numbers)

$N = \{0,0,0\}\{0,0,1\}\{0,0,2\}$

$T =$

$\tau =$ if ($C_{c+2}[t]$ - soil mobility) < $C_{c+1}[t]$

$Cc[t+q] = Cc[t] - (C_{c+2}[t]$ - soil mobility)

else C

$Cc[t+q] = Cc[t] - C_{c+1}[t]$

$qZ_0^+ = 1$ second

Soil Depth (plane 1):

$CA = \langle S,n,C,N,T,\tau,qZ_0^+\rangle$

$S = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$.

$n = (1,1,2)$

$C = 0.0 \rightarrow \infty$ (all positive real numbers)

$N = \{0,0,0\}\{0,0,1\}$

$T =$

$\tau =$ if ($C_{c+1}[t]$ - soil mobility) < $Cc[t]$

$Cc[t+q] = Cc[t] - (C_{c+1}[t]$ - soil mobility)

else C

$Cc[t+q] = Cc[t] - Cc[t]$

$qZ_0^+ = 1$ second

Energy(plane 2):

$CA = \langle S,n,C,N,T,\tau,qZ_0^+\rangle$

$S = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$.

$n = (3,3,12)$

$C = 0.0 \rightarrow \infty$ (all positive real numbers)

$N =$ $\{0,0,-1\}$

$\{-1,1,0\}\{0,1,0\}\{1,1,0\}$

$\{-1,0,0\}\{0,0,0\}\{1,0,0\}$

$\{-1,-1,0\}\{0,-1,0\}\{1,-1,0\}$

... The neighborhood is square from relative plane 0 to relative plane 11

$T =$

$\tau =$ if ($C_C[t]$ - soil mobility) < $C_{c+(0,0,-1)}[t]$

D =($Cc[t]$ - soil mobility)

else

$D = C_{c+(0,0,-1)}[t]$

if ((k* $Cc[t] / C_{c+(0,0,1)}[t]$) - p_runup_loss) > $C_{c+(0,0,1)}[t]$

R = p_runup_loss

else

R = (k* $Cc[t] / C_{c+(0,0,1)}[t]) - C_{c+(0,0,1)}[t]$

$Cc[t+q] = (Cc[t] + Cc[t]^* D / C_{c+(0,0,1)}[t]) + (k^*C_{c+(0,0,1)}[t]^*D + D^*D)) +$

$(Cc[t] - k * R * C_{c+(0,0,1)}[t]) +$
$(C_{c+(0,0,1)}[t] - (C_{c+(0,0,2)}[t] + C_{c+(0,0,3)}[t] + C_{c+(0,0,4)}[t] + C_{c+(0,0,5)}[t] +$
$C_{c+(0,0,6)}[t] + C_{c+(0,0,7)}[t] + C_{c+(0,0,8)}[t] + C_{c+(0,0,9)}[t])) *$
$Cc[t]/ C_{c+(0,0,1)}[t] + (C_{c+(1,1,7)}[t] * C_{c+(1,1,7)}[t] / C_{c+(1,1,7)}[t] + C_{c+(0,1,8)}[t] *$
$C_{c+(0,1,8)}[t] / C_{c+(0,1,8)}[t] + C_{c+(-1,1,9)}[t] * C_{c+(-1,1,9)}[t] / C_{c+(-1,1,9)}[t] +$
$C_{c+(-1,0,2)}[t] * C_{c+(-1,0,2)}[t] / C_{c+(-1,0,2)}[t] + C_{c+(-1,-1,3)}[t] * C_{c+(-1,-1,3)}[t] /$
$C_{c+(-1,-1,3)}[t] + C_{c+(0,-1,4)}[t] * C_{c+(0,-1,4)}[t] / C_{c+(0,-1,4)}[t] + C_{c+(1,-1,5)}[t] *$
$C_{c+(1,-1,5)}[t] / C_{c+(1,-1,5)}[t] + C_{c+(1,0,6)}[t] * C_{c+(1,0,6)}[t] / C_{c+(1,0,6)}[t])$

qZ_0^+ = 1 second

Debris Thickness (plane 3):

CA = <S,n,C,N,T,τ,qZ_0^+>

S = 0 1 2 3 4 5 6 7 8 9 .
n = (3,3,11)
C = 0.0 -> ∞ (all positive real numbers)
N = {0,0,-2}
{-1,1,-1} {0,1,-1} {1,1,-1}
{-1,0,-1} {0,0,-1} {1,0,-1}
{-1,-1,-1} {0,-1,-1} {1,-1,-1}
... planes 0 through 8 are also square neighborhoods
T =
τ = if ($C_{c+(0,0,-1)}[t]$ - soil mobility) < $C_{c+(0,0,-2)}[t]$
D =($C_{c+(0,0,-1)}[t]$ - soil mobility)
else
D =$C_{c+(0,0,-2)}[t]$
$Cc[t+q] = Cc[t+q] + D + SUM[i=1..9](C_{c+(x,y,i)}[t] - C_{c+i}[t])$

The full expansion of this rule is in Landslide.ma and is too large and complex to expand here. It is: subtract the outflow to cell i from the inflow from cell i and add that to the new thickness

qZ_0^+ = 1 second

Debris Outflow (planes 4-11):

Written for plane 4, decrement all z values by n for other planes, where n = plane # - 4)

CA = <S,n,C,N,T,τ,qZ_0^+>

S = 0 1 2 3 4 5 6 7 8 9 .
n = (3,3,13)
C = 0.0 -> ∞ (all positive real numbers)
N = {-1,1,-4} {0,1,-4} {1,1-4}
{-1,0,-4} {0,0,-4} {1,0,-4}
{-1,-1,-4} {0,-1,-4} {1,-1,-4}
... all other planes are identical
T =
τ = Where x&y are (1,1) for plane 4, (0,1) for plane 5, (-1,1) for plane 6,
(-1,0) for plane 7, (-1,-1) for plane 8, (0,-1) for plane 9, (1,-1) for plane 10,
and (1,0) for plane 11.

if (C[t] + C[t]) > pf) AND Average Debris Flow > ($C_{c+(x,y,-4)}$[t] + $C_{c+(x,y,-1)}$[t])
Cc[t+q] = Average Debris Flow - ($C_{c+(x,y,-4)}$[t] + $C_{c+(x,y,-1)}$[t])
else
Cc[t+q] = 0
qZ_0^+ = 1 second

Runnup (plane 12):

CA = <S,n,C,N,T,τ,qZ_0^+>
S = 0 1 2 3 4 5 6 7 8 9 .
n = (1,1,12)
C = 0.0 -> ∞ (all positive real numbers)
N = {0,0,-11} {0,0,-10}...{0,0,0}
T =
τ = if ($C_{c+(0,0,-10)}$[t] - soil mobility) < $C_{c+(0,0,-11)}$[t]
D =($C_{c+(0,0,-10)}$[t] - soil mobility)
else
D =$C_{c+(0,0,-11)}$[t]
Cc[t+q] = Cc[t] + D
qZ_0^+ = 1 second

Simulation Report

Using the same 10x10 map, shown below, 5 different inital conditions were created, using the .val files. In each case there is 4 units (meters) of erodable soil in each cell. These are run with the batch files:

- nothing.bat — Nothing should happen, no energy in the system.
- peaks.bat — Two energy and debris initial values at the peaks at (0,0) and (9,0)
- peaks_high.bat — Same as peaks, but more energy and debris
- two_ways.bat — Energy and debris at (5,8) and (5,9)
- low.bat — Energy and debris at (0,9) and (9,9)

Each of these .bat files as a coressponding _drw.bat which is used by drawlog to generate the draw files. Each .bat file has a matching .val file as well, with the correct starting values.

Altitude Map:

60	50	40	50	60	50	40	30	20	10
50	50	40	50	60	40	40	30	20	10
50	40	40	40	50	40	30	30	20	10
40	40	40	40	40	40	30	40	40	40
40	40	40	30	30	30	40	50	50	50
40	40	40	30	30	40	50	50	60	60
50	40	40	40	30	40	40	50	50	50
50	50	50	40	30	30	30	40	40	40

60	60	60	50	50	30	30	30	20	20
70	70	70	60	60	50	50	30	20	10

Nothing.bat

Correctly, nothing happens, there is no energy and no debris in the system, therefore nothing happens. The output files nothing_test.log and nothing_test.drw.

This also generates the output files parse.txt and debug.txt. It is intended to be the simplest possible test to confirm that the basic rules of the model are written correctly.

Peaks.bat

The slides start well enough but soon stop, even though there is flat space, or a slope for them to spread out on. This may be due to the energy involved or the SCIDDICA paramters which I do not have confidence in as they are arbitrary and chosen to get the simulation running.

The slide starting at (9,0) travels farther, but this is likely because it does not run into level ground for longer.

Peaks_high.bat

Displays identical behavior to Peaks, but with more debris, as the starting debris amount was greater. Indicates that the problem with slides on flat ground is probably not the kinetic energy available.

Two_ways.bat

Speards out as expected, fills in all the lower lying areas with out spreading upwards when it reaches higher ground.

Low.bat

Energy and debris are not thick enough to spill up onto the higher areas surrounding the low places. However, the debris doesn't speard out to cover level ground.

Conclusion

While the basic model seems to be working there are problems with slides on level ground not spreading in the expected manner. The initial parameters may be at fault for this as they were chosen arbitrarily when no test values could be found.

Also the model is not as accurate as it could be because implementing the entirety of the SCIDDICA outflow rules requires looping over the rules several times. Cell-DEVS cannot handle this.

References

[1] Calidonna, C. R., Di Napoli, C., Giordano, M., Furnari, M. M., Di Gregorio, S., “A Network of Cellular Automata for a Landslide Simulation”, *Proceedings of the 15th international conference on Supercomputing ICS '01*, 2001.

[2] D'Ambrosio, D., Di Gregorio, S., Iovine, G., “Simulating debris flows through a hexagonal cellular automata: SCIDDICA $S_{3\text{-hex}}$”, *Natural Hazards and Earth System Sciences*, vol 3. pp. 545-559, 2003.

[3] Di Gregorio, S., Rongo, R., Siciliano, C., Sorriso-Valvo, M., Spataro, W., “Mount Ontake Landslide Simulation by the Cellular Automata Model SCIDDICA-3”, *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, vol. 24 no. 2, pp. 131-137, 1999.