A SHORT INTRODUCTION TO LANDSCAPE EVOLUTION MODELING

GREGORY E. TUCKER, UNIVERSITY OF COLORADO

CONTENTS

1. Overview

The learning goals of this section of the short course are:

- To gain a clearer understanding of how a typical landscape evolution model (LEM) solves the type of governing equations that we saw on Day 1.
- To understand how continuity of mass is maintained by a typical LEM, and some of the limitations that arise.
- To appreciate some of the ways in which climate and hydrology can be represented in a LEM, and some of the simplifications involved.
- To appreciate that working with LEMs involves choosing a level of simplification in the governing physics that is appropriate to the problem at hand.
- To gain hands-on experience actually using a LEM.
- To get a sense for how and why hillslope diffusion produces convex hillslopes.
- To appreciate the concepts of transient versus steady topography.
- To acquire a feel for the similarity and difference between detachment-limited and transport-limited modes of fluvial erosion.
- To understand the connection between fluvial physics and slope-area plots.
- To appreciate that LEMs (1) are able to reproduce (and therefore, at least potentially, explain) common forms in fluvially carved landscapes, (2) can enhance our insight into dynamics via visualization and experimentation, but (3) leave open many important questions regarding long-term process physics.
- To develop a sense for how to use landscape evolution models.

2. Introduction to LEMs

2.1. **Brief History.**

- G.K. Gilbert's "word pictures" (Gilbert, 1877)

TABLE 1. Partial list of numerical landscape models published between 1991 and 2005.

Model	Example reference	Notes
SIBERIA	Willgoose et al. (1991)	Transport-limited; Channel activator function
DRAINAL	Beaumont et al. (1992)	"Undercapacity" concept
GILBERT	Chase (1992)	Precipiton
DELIM/MARSSIM	Howard (1994)	Detachment-limited; Nonlinear diffusion
GOLEM	Tucker and Slingerland (1994)	Regolith generation; Threshold landsliding
CASCADE	Braun and Sambridge (1997)	Irregular discretization
CAESAR	Coulthard et al. (1996)	Cellular automaton algorithm for 2D flow field
ZSCAPE	Densmore et al. (1998)	Stochastic bedrock landsliding algorithm
CHILD	Tucker and Bras (2000)	Stochastic rainfall
EROS	Crave and Davy (2001)	Modified precipiton
TISC	Garcia-Castellanos (2002)	Thrust stacking
LAPSUS	Schoorl et al. (2002)	Multiple flow directions
APERO/CIDRE	Carretier and Lucazeau (2005)	Single or multiple flow directions

- 1960s: 1D models (Culling, Ahnert, Kirkby, Young)
- 1970s: 2D models
- Late 1980s and 1990s: beginning of "modern era," (SIBERIA, GILBERT, DRAINAL, followed by others).
- Today: many model codes (see Table 1 for a partial list), many applications, scales, and objectives, ranging from soil erosion to continental collision.

2.2. Brief Overview of Models and their Uses.

- Some examples: coupled erosion-deposition systems, meandering, Mars cratering, forecasting mine-spoil degradation, estimating erosion risk to buried hazardous waste.
- Overview: powerful tools, but with process ingredients that are provisional and subject to test. Important to have continuing cross-talk between models and observations (subjects for later in the week).
- This morning: overview of how a LEM works, including how terrain and water flow are represented numerically and how the processes we reviewed yesterday are computed.

3. Continuity of Mass and Discretization

Yesterday we saw examples of mass continuity equations for a column of soil or rock, such as:

$$(1) \qquad \frac{\partial \eta}{\partial t} = B - \nabla \vec{q}_s$$

This is one of several variations; for discussion of others, see Tucker and Hancock (2010). Some models, for example, distinguish between a regolith layer and the bedrock underneath. [illustration of this] Note that this type of mass continuity equation applies only to terrain that has one and only one surface point for each coordinate; not, for example, a vertical cliff or an overhang.

A LEM computes $\eta(x, y, t)$ given (1) process rules, (2) initial conditions, and (3) boundary conditions.

One thing all LEMs have in common is that they divide the terrain into discrete elements. Often these are square elements, but not always (CASCADE and CHILD). [Illustration of an element]

For a discrete element, continuity of mass enforced by the following equation (in words):

Time rate of change of mass in element = mass rate in at boundaries - mass rate out at boundaries + inputs or outputs from above or below (tectonics, dust deposition, etc.)

Mathematically,

$$(2) \qquad \frac{d\eta_i}{dt} = B + \frac{1}{\Lambda_i} \sum_{j=1}^{N} q_{sj} \lambda_j$$

Notice that, for the sake of simplicity, I've pulled a trick here in writing this in terms of *volumetric* fluxes. This is known as a *finite-volume* method because it is based on computing fluxes in and out along the boundaries of a finite volume of space.

Some terminology: a *cell* is a patch of ground with boundaries called *faces*. A *node* is the point inside a cell at which we track elevation (and other properties). On a raster grid, cells are square and nodes fall at the center of the cell. On the irregular mesh used by CASCADE and CHILD, the *cell* is the area of land that is closer to that particular node than to any other node in the mesh. (It is a mathematical entity known as a *Voronoi cell* or *Thiessen polygon*; for more, see Braun and Sambridge (1997), Tucker et al. (2001))

Equation 2 gives us the time derivatives of every node on the grid. How do we solve for the new elevations at time t? There are many ways to do this, including matrix-based implicit solvers (see for example Fagherazzi et al. (2002)). We won't

get into the details of numerical solutions (at least not yet), but for now note that the simplest solution is the forward-difference approximation:

$$(3) \qquad \frac{d\eta_i}{dt} \approx \frac{\eta_i(t + \Delta t) - \eta_i(t)}{\Delta t}$$

$$(4) \qquad \eta_i(t + \Delta t) = \eta_i(t) + U\Delta t + \Delta t \frac{1}{\Lambda_i} \sum_{j=1}^{N} q_{sj}\lambda_j$$

The main disadvantage of this approach is that very small time steps are typically needed in order to ensure numerical stability. (CHILD uses a variant of this that seeks the smallest possible value of Δt at each iteration). A good discussion of numerical stability, accuracy, and alternative methods for diffusion-like problems can be found in Press et al. (2007).

4. Gravitational Hillslope Transport

Geomorphologists often distinguish between hillslope and channel processes. It's a useful distinction, although one has to bear in mind that the transition is not always abrupt, and even where it is abrupt, it is often either discontinuous or highly dynamic or both.

Alternatively, one can also distinguish between processes that are driven nearly exclusively by gravitational processes, and those that involve a fluid phase (normally water or ice). This distinction too has a gray zone: landslides are gravitational phenomena but often triggered by fluid pore pressure, while debris flows are surges of mixed fluid and solid. Nonetheless, we will start with a consideration of one form of gravitational transport on hillslopes: soil creep.

As noted on Day 1, for relatively gentle, soil-mantled slopes, there is reasonably strong support for a transport law of the form:

$$(5) \qquad \vec{q}_s = -D\nabla\eta$$

Using the finite-volume method outlined in equation 2, we want to calculate $\vec{q}_s$ at each of the cell faces. Suppose node i and node k are neighboring nodes that share a common face (we'll call this face j). We approximate the gradient between nodes i and k as:

$$(6) \qquad q_{sik} \simeq -D\frac{\eta_k - \eta_i}{L_{ik}}$$

where q_{sik} is the volume flux per unit width from node k to node i (if negative, sediment flows from i to k), and L_{ik} is the distance between nodes. On a raster grid, $L_{ik} = \Delta x$ is simply the grid spacing. To compute the total sediment flux through

face j, we simply multiply the unit flux by the width of face j, which we denote λ_{ij} (read as "the j-th face of cell i):

$$(7) \qquad\qquad\qquad Q_{sik} = q_{sik}\lambda_{ij}$$

[Illustration of cells and fluxes]

Getting Set Up with CHILD. Our first exercise is simply to run the executable file to make sure it is installed and working correctly. In some cases, it might be necessary to create a new executable file from the source code. For these exercises, the executables should be in folders under `ChildExercises/Executables`, with folders for `IntelMacOS10.6`, `IntelMacOS10.5andEarlier`, `Cygwin` and `Win32`. For convenience, copy the binary for your platform from its subfolder to the `ChildExercises` folder.

To navigate folders and run CHILD, you will need to use a UNIX shell (unless working in the Command window under Windows): either the **Terminal** application on a Mac, or a UNIX emulator shell such as cygwin on a PC. To change folders ("directories" in UNIX-speak), use `cd` followed by the folder name. A single period represents the current working directory; two periods represent the next directory up. For example, the command `cd ..` takes you one level up. To get a list of files in a directory, use `ls`. (For Command prompt under windows, use `dir` instead of `ls` and backslashes instead of forward slashes).

On a Mac: create a new terminal window by running the Terminal application (usually found under Applications/Utilities). Navigate to the folder containing the executable file `child` and type `./child`.

On a Linux computer: create a new unix terminal window. Navigate to the folder containing the executable file `child` and type `./child`.

On a PC with Cygwin: start up cygwin. In the cygwin command window, navigate to the folder containing the executable file `child.exe` and type `./child`.

On a PC: start up Command Window. In the command window, navigate to the folder containing the executable file `child.exe` and type `child`.

You should see something like the following:

```
Usage: ./child [options] <input file>
 --help: display this help message.
 --no-check: disable CheckMeshConsistency().
 --silent-mode: silent mode.
 --version: display version.
```

While we're at it, let's get ready to visualize the output. Start Matlab. The first thing we will do is tell Matlab where to look for the plotting programs that we will use. At the Matlab command prompt type:

`path( path, '`*childFolderLocation*`/ChildExercises/MatlabScripts' )`

For *childFolderLocation*, use the path name of the folder that contains the CHILD package.

In Matlab, navigate the current folder to the location of the example input file `hillslope1.in` (which should end in: `ChildExercises/Hillslope1`).

(Windows note: use a backslash instead of a forward slash).

CHILD Exercise: Hillslope Diffusion and Parabolic Slopes.

(1) *In your terminal window, navigate to the* `ChildExercises/Hillslope1` *folder.*

(2) *To run the example, in your terminal window type:*
   `../child hillslope1.in`

(3) *A series of numbers will flash by on the screen. These numbers represent time intervals in years. The 2-million-year run takes about 20 seconds on a 2GHz Intel Mac. When it finishes, return to Matlab and type:*
   `m = cmovie( 'hillslope1', 21, 200, 200, 100, 50 );`
   (This command says "generate a 21-frame movie from the run 'hillslope1' with the x-, y- and z- axes set to 200, 200 and 100m, respectively, and with the color range representing 0 to 50m elevation).

(4) *To replay the movie, type* `movie(m)`*.*

(Windows note: we found that under Vista and Windows 7, the movie figure gets erased after display; slightly re-sizing the figure window seems to fix this).

The analytical solution to elevation as a function of cross-ridge distance y is:

$$(8) \qquad z(y) = \frac{u}{2D} \left(L^2 - (y - y_0)^2 \right)$$

where L is the half-width of the ridge (100m in this case) and $y0$ is the position of the ridge crest (also 100m). The effective uplift rate u, represented in the input file by the parameter `UPRATE`, is $10^{-4} m/yr$. The diffusivity coefficient D, represented in the input file by parameter `KD`, is $0.01 \text{m}^2/\text{yr}$. Next, we'll make a plot that compares the computed and analytical solutions.

Type the following in Matlab:

(1) `ya = 0:200;`   *This is our x-coordinate*

(2) `u = 0.0001; D = 0.01; y0 = 100; L = 100;`

(3) `za = (u/(2*D))*(L^2-(ya-y0).^2);`

(4) `figure(2), plot( ya, za ), hold on`

(5) `xyz = creadxyz( 'hillslope1', 21 );`   *Reads node coords, time 21*

(6) `plot( xyz(:,2), xyz(:,3), 'r.' ), hold off`

(7) `legend( 'Analytical solution', 'CHILD Nodes' )`

Diffusion theory predicts that equilibrium height varies linearly with u, inversely with D, and as the square of L. Make a copy of `hillslope1.in` *and open the copy in a text editor. Change one of these three parameters. To change u, edit the number below the line that starts with* `UPRATE`. *Similarly, to change D, edit the value of parameter* `KD`. *If you want to try a different ridge width L, change both* `Y_GRID_SIZE` *and* `GRID_SPACING` *by the same proportion (changing* `GRID_SPACING` *will ensure that you keep the same number of model nodes). Re-run CHILD with your modified input file and see what happens.*

4.1. **Nonlinear Diffusion.** As discussed on Day 1, the simple slope-linear transport law works poorly for slopes that begin to approach the angle of repose for sediment and rock. The next example explores what happens to our ridge when we (1) increase the relative uplift rate, and (2) use the nonlinear diffusion transport law:

$$(9) \qquad \vec{q_s} = \frac{-D\nabla z}{1 - |\nabla z/S_c|^2}$$

Nonlinear Diffusion and Planar Slopes.
  (1) *Navigate to the* `Hillslope2` *folder*
  (2) *Run CHILD:* `../child hillslope2.in`
  (3) *In Matlab, navigate to the* `Hillslope2` *folder*
  (4) *When the 70,000-year run (∼1 minute on a 2GHz mac) finishes, type in Matlab:*
    `m = cmovie( 'hillslope2', 21, 200, 200, 100, 70 );`

If we had used linear diffusion, the equilibrium slope gradient along the edges of the ridge would be $S = uL/D = (0.001)(100)/(0.01) = 10m/m$, or about 84°! Instead, the actual computed gradient is close to the threshold limit of 0.7. Notice too how the model solution speed slowed down as the run went on. This reflects the need for especially small time steps when the slopes are close to the threshold angle.

4.2. **Remarks.** There is a lot more to mass movement than what is encoded in these simple diffusion-like transport laws. Some models include stochastic landsliding algorithms (e.g., CASCADE, ZSCAPE). Some impose threshold slopes (e.g., GOLEM). One spinoff version of CHILD even includes debris-flow generation and routing (Lancaster et al., 2003).

5. Rainfall, Runoff, and Drainage Networks

In order to calculate erosion, sediment transport, and deposition by running water, a model needs to know how much surface water is flowing through each cell in the

model. Usually, the erosion/transport equations require either the total discharge, Q [L^3/T], the discharge per unit channel width, q [L^2/T], or the flow depth, H.

There are three main alternative methods for modeling the flow of water across the landscape:

(1) Methods based on contributing drainage area
(2) Numerical solutions to the 2D, vertically integrated and time-averaged Navier-Stokes equations
(3) Cellular automaton methods

5.1. **Methods Based on Drainage Area.** On Day 1 we were introduced to the concept of *drainage area*: the horizontally projected area of land that contributes flow through to a particular channel cross-section or to unit length of contour on a hillslope.

Quick refresher: A is the size of the land area that contributes flow to a particular cross-section of stream channel. A/b is the area per unit contour length (b is contour length) contributing flow to a particular point on a hillslope.

For a numerical landscape model that uses discrete cells, A is defined as the area that contributes flow to a particular cell. When topography is represented as a raster grid, the most common method for computing drainage area is the *D8 method*. Each cell is assigned a flow direction toward one of its 8 surrounding neighbors. An algorithm is then used to trace flow paths downstream and add up the number of cells that contribute flow each cell.

[D8 illustration]

[Simple 3x3 example of flow directions and drainage areas on a raster grid]

For the Voronoi cell matrix that CHILD and CASCADE use, the simplest routing procedure is a generalization of D8. Each cell i has N_i neighbors. As we noted earlier, the slope from cell i to neighbor cell k is defined as the elevation difference between the nodes divided by the horizontal distance between them. Thus, one can define a slope for every *edge* that connects each pair of nodes. There is a slope value for each of the N_i neighbors of node i. The flow direction is assigned as the steepest of these slopes.

[Voronoi routing illustration]

Single-direction flow algorithms have advantages and disadvantages. Some models use a *multiple flow direction* approach to represent the divergence of flow on relatively gentle slopes or divergent landforms. This is most appropriate for models that operate on a grid resolution significantly smaller than the length of a hillslope. When grid cells are relatively large, conceptually each cell contains a primary channel, narrower than the cell, that is tracked.

Flow Over Noisy, Inclined Topography.

(1) *In the terminal window, navigate to the* `Network1` *folder and run the input file by typing:*

```
../child network1.in
```

(2) *In Matlab, navigate to the* `Network1` *folder*
(3) `figure(1), clf`
(4) `colormap pink`
(5) `a = cread( 'network1.area', 1 );`
(6) `ctrisurf( 'network1', 1, a );`
(7) `view( 0, 90 ), shading interp, axis equal`

The networks are formed because of noise ($\pm 1m$ in this case) in the initial surface, which causes flow to converge in some places.

The simplest method for computing discharge from drainage area is to simply assume (1) all rain runs off, and (2) rain lasts long enough that the entire drainage network is in hydrologic steady state. In this case, and if precipitation rate P is uniform,

$$(10) \qquad\qquad\qquad\qquad Q = PA$$

A number landscape modeling studies have used this assumption, on the basis of its simplicity, even though it tends to make hydrologists faint! The simplicity is indeed a virtue, but one needs to be extremely careful in using this equation, for at least three reasons. First, obviously Q varies substantially over time in response to changing seasons, floods, droughts, etc. We will return to this issue shortly. Second, there is probably no drainage basin on earth, bigger than a hectare or so, from which *all* precipitation runs off. Typically, evapotranspiration returns more than half of incoming precipitation to the atmosphere. Third, hydrologic steady state is rare and tends to occur only in small basins, though it may be a reasonable approximation for mean annual discharge in some basins.

[Susquehanna and/or Kentucky River basin examples]

River discharge, whether defined as mean annual, bankfull, mean peak, or some other way, often shows a power-law-like correlation with drainage area. Some models take advantage of this fact by computing discharge using an empirical approach:

$$(11) \qquad\qquad\qquad\qquad Q = bA^c$$

where c typically ranges from 0.5-1 and b is a runoff coefficient with awkward units that represents a long-term "effective" precipitation regime.

CHILD's default method for computing discharge during a storm takes runoff at each cell to be the difference between storm rainfall intensity P and soil infiltration

capacity I:

$$(12) \qquad\qquad Q = (P - I)A$$

which of course is taken to be zero when $P < I$.

5.2. **Shallow-Water Equations.** Some landscape models are designed to address relatively small-scale problems such as channel initiation, inundation of alluvial fan surfaces, channel flood flow, etc. In such cases, the convergence and divergence of water in response to evolving topography is an important component of the problem, and is not adequately captured by the simple routing schemes described above. Instead, a tempting tool of choice is some form of the *shallow-water equations*, which are the vertically integrated form of the general (time-averaged) viscous fluid-flow equations. One form of the full shallow-water equations is:

$$(13) \qquad \frac{\partial \eta}{\partial t} = i - \left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} \right)$$

$$(14) \qquad \frac{\partial q_x}{\partial t} + \frac{\partial q_x u}{\partial x} + \frac{\partial q_y u}{\partial y} + gh\frac{\partial h}{\partial x} + gh\frac{\partial \eta}{\partial x} + \frac{\tau_{bx}}{\rho} = 0$$

$$(15) \qquad \frac{\partial q_y}{\partial t} + \frac{\partial q_y v}{\partial y} + \frac{\partial q_x v}{\partial x} + gh\frac{\partial h}{\partial y} + gh\frac{\partial \eta}{\partial y} + \frac{\tau_{by}}{\rho} = 0$$

These equations express continuity of mass, x-directed momentum, and y-directed momentum, respectively. They are challenging and computationally expensive to integrate numerically in their full form. However, there are several approximate forms that are commonly used, including the non-accelerating flow form (in which convective accelerations are assumed negligible) and the kinematic-wave equations (in which gravitational and friction forces are assumed to dominate). An example of use of the shallow-water equations in a landform evolution model can be found in the work of T.R. Smith and colleagues. Various forms of the shallow-water equations can often be found in hydrologic models, and sometimes in soil-erosion models (e.g., Mitas and Mitasova, 1998).

[Bedford et al. rill-and-pond simulation example]

5.3. **Cellular Automata.** Some models use cellular automaton methods to calculate flow over a cellular topography. These include:

- Chase's precipiton algorithm
- Crave-Davy modified precipiton algorithm
- Murray-Paola multiple-flow-direction river-flow algorithm
- Coulthard's generalization of Murray-Paola for 2D flow (CAESAR model)

5.4. **Depressions in the Terrain.** What happens when flow enters a topographic depression? In the real world, three possibilities: complete evaporation/infiltration, formation of a lake with overflow, or formation of a closed lake.

CHILD uses a lake-fill algorithm. [Animation example showing lake-fill algorithm used to simulate fan-deltas in a subsiding marine basin]

5.5. **Precipitation and Discharge.**
- Water supply to channel network varies dramatically in both time and space
- Big gap in time scale between storms/floods and topographic evolution
- "Effective discharge" concept: discharge that represents the cumulative geomorphic effect of the natural sequence of storms and floods
- Willgoose et al. (1991): use mean peak discharge
- Huang and Niemann (2006): effective discharge not necessarily the same at different times and places

Basically, landscape models tend to use one of four methods:

(1) Steady flow, uniform precipitation or runoff coefficient (effective discharge concept)
(2) Steady flow, nonuniform precipitation/runoff (e.g., orographic precipitation)
(3) Stochastic-in-time, spatially uniform runoff generation
(4) "Short storms" model (Sólyom and Tucker, 2004)

We do not have time to look in detail at all of these. Instead, we will take a brief look at the Poisson rectangular pulse model implemented in CHILD.

Visualizing a Poisson Storm Sequence.

(1) *In the terminal window, navigate to the* `Rainfall1` *folder and run the input file by typing:*
```
../child rainfall1.in
```
(2) *In Matlab, navigate to the* `Rainfall1` *folder*
(3) *In Matlab:* `figure(1), clf, cstormplot( 'rainfall1' );`
(4) `figure(2), clf, cstormplot( 'rainfall1', 10 );`

The first plot shows a 1-year simulated storm sequence; the second shows just the first 10 storms.

The motivation for using a stochastic flow model is (1) that nature *is* effectively stochastic, and (2) variability matters when the erosion or transport rate is a nonlinear function of flow. For more on this, see Tucker and Bras (2000); Snyder et al. (2003); Tucker (2004), and DiBiase et al. (2009).

5.6. **Remarks.** Landscape evolution models can be, and have been, used to study climate impacts on erosion, topography, and mountain building. But be careful— climate and hydrology amount to much more than a "sprinkler over the landscape."

6. Hydraulic Geometry

- Channel size, shape, and roughness controls delivery of hydraulic force to the bed and banks
- Most landscape models either implicitly assume constant width (practical but dangerous) or use the empirical relation $W = K_w Q^b$, where $b \approx 0.5$.
- Models with time-varying discharge must also specify how width varies at a point along the channel as Q rises and falls.
- Width-discharge scaling is practical but incomplete, because we observe that channels can narrow or widen downstream in concert with variations in incision rate.
- Examples from Taiwan, Italy.
- Model examples from Wobus et al. (2006), Attal et al. (2008).
- Full treatment of the channel geometry adjustment problem is a frontier area.

7. Erosion and Transport by Running Water

- Several competing models for erosion by channelized flow
- Detachment-limited models: assume that eroded material leaves the system without significant re-deposition; lowering of channel limited by detachment
- Transport-limited models: assume plentiful supply of loose sediment; lowering of channel limited capacity to transport sediment
- Simple hybrid: lowering may be limited either by excess transport capacity by detachment rate, depending on local sediment supply and substrate resistance
- Undercapacity concept: detachment rate depends on surplus transport capacity
- Saltation-abrasion model: detachment driven by grain impacts, limited by sediment shielding

7.1. **Detachment-Limited Models.** On a bed with little loose sediment, detachment of particles may be driven primarily by hydraulic lift and drag ("plucking"). Most models assume that the rate of detachment (or more generally the capacity for detachment) depends on excess bed shear stress:

$$(16) \qquad D_c = K_b \left(\tau - \tau_c \right)^{p_b}$$

where τ is local bed shear stress, τ_c is a threshold stress below which detachment is ineffective, K_b is a constant, and p_b is an exponent.

Bed shear stress fluctuates in space and time, but is often treated using the cross-sectional average, which in turn is based on a force balance between gravity and friction.

Some models assume detachment rate depends on stream power per unit width $\omega = (Q/W)S$:

$$(17) \qquad D_c = K_b \left(\frac{Q}{W} S - \Phi_c \right)$$

Stream power per unit width turns out to be proportional to $\tau^{3/2}$, so the two erosion formulas are closely related (Whipple and Tucker, 1999). In the following example, we will use the unit stream power formula with $\Phi_c = 0$.

Detachment-Limited Hills and Mountains.

(1) *In the terminal window, navigate to the* `Dlim` *folder and run the input file by typing:*
    `../child dlim.in`
    The 3m.y. run should take about 20 seconds.
(2) *In Matlab, navigate to the* `Dlim` *folder*
(3) *In Matlab:* `figure(1), clf, colormap jet`
(4) `cmovie( 'dlim', 31, 3e4, 3e4, 1e3, 500 );`
(5) `figure(2), clf`
(6) `csa( 'dlim', 31 );`   *Shows slope-area graph*

Notice that the landscape has come close to a state of equilibrium between erosion and relative uplift. The resulting terrain has about 200m of relief over a 30km half-width mountain range—more Appalachian than Himalayan. Notice that the log-log slope-area graph shows a straight line, indicating a power-law relationship. This is exactly to be expected, and we can predict the plot slope and intercept analytically. Finally, note the points on the upper left of the graph. These "first order" cells, about 2500m^2 in area, have slopes less than 10%. They represent embedded channels, not hillslopes, which are too small to resolve at this grid spacing.

Now, what happens when we increase the relative uplift rate?

(1) *Run the* `dlimC1.in` *input file by typing:*
    `../child dlimC1.in`
    This run starts off where the previous one ended, but with a 10x higher rate of relative uplift.
(2) *In Matlab:* `figure(1)`
(3) `cmovie( 'dlimC1', 31, 3e4, 3e4, 1e4, 5000 );`   *10x vertical scale*
(4) `figure(2), clf`
(5) `hold on, csa( 'dlimC1', 31, 'r.'  ); hold off`

Because we are using a slope-linear detachment law, a 10x increase in relative uplift rate leads to a 10x increase in relief. Notice that the points have shifted upward by a factor of 10 on the slope-area graph.

We still do not see any hillslopes, because the scale of landscape dissection is too fine for the model to resolve. Next, we will "zoom in" by repeating the `dlim` run but with a twenty-fold decrease in domain size and model cell size.

Zooming in to the Hillslopes.

(1) *Run the* `dlim_small.in` *input file by typing:*

```
../child dlim_small.in
```
   This run is identical to `dlim` *but with a domain of 1.5 by 1.5km and ∼25m wide cells, instead of 30x30km and ∼500m cells.*

(2) *In Matlab:* `figure(1)`

(3) `cmovie( 'dlim_small', 31, 1.5e3, 1.5e4, 500, 200 );`

(4) `figure(2), clf`

(5) `hold on, csa( 'dlim_small', 31, 'g.'  ); hold off`

Note how the hillslopes become evident in the topography. In the slope-area plot, the points seem to continue the trend of the coarser-scale run, but somewhat shifted upward. Can you guess why they are shifted upward? (The answer is subtle, and lies hidden in `dlim_small2.in`).

For the next exercise, we return to our earlier `dlimC1` run and plot a representative stream profile at different times, to look at how the profile responds to the increased rate of relative uplift.

Knickzones and Transient Response.

(1) *In Matlab:* `figure(1), clf`

(2) `[d,h,x,y] = cstrmproseries( 'dlimC1', 10, 15000, 29000 );`
   This command traces the stream profile starting from $x = 15km$, $y = 29km$. *It will plot the first 10 profiles.*

(3) `figure(2), clf, plot( x, y )`
   This shows the horizontal trace of the stream course.

During the period of transient response, the stream profile shows a pronounced convexity, or knickzone, along the profile. The knickzone marches upstream through time. This pattern is characteristic of the "stream power" erosion law, which is actually a form of wave equation.

7.2. **Transport-Limited Models.** We next explore the dynamics of landscapes and networks with transport-limited models. One caution as we do so: we will assume that channel width is independent of grain size, slope, etc.

A Pile of Fine Sand.

(1) *In the terminal window, navigate to the* `Tlim` *folder and run:*

```
../child tlim1.in
```
   The 1m.y. run should take about 2 minutes.

(2) *In Matlab, navigate to the* `Tlim` *folder*
(3) *In Matlab:* `figure(1), clf`
(4) `cmovie( 'tlim1', 21, 3e4, 3e4, 40, 10 );`
(5) `figure(2), clf`
(6) `csa( 'tlim1', 21 ); axis([1e-1 1e3 1e-4 1e-3])`

In this run, we are effectively assuming that 0.1mm sand moves as bed-load, according to a Meyer-Peter and Mueller-like transport formula. The landscape takes on an effectively uniform and very shallow gradient, on the order of 3×10^{-4}.

Now let's try the same experiment with 5cm cobbles.

A Pile of Cobbles.

(1) *Run:*
    `../child tlim2.in`
    The 3m.y. run should take about 2-3 minutes.
(2) *In Matlab:* `figure(1), clf`
(3) `cmovie( 'tlim2', 21, 3e4, 3e4, 1000, 300 );`
(4) `figure(2)`
(5) `hold on, csa( 'tlim2', 31, 'r.'  ); hold off`

Lesson: grain size matters!

But let's remember the caveat that channel width matters too, and we haven't taken that into account with these simple runs. Also, Nicole Gasparini's work (Gasparini et al., 1999, 2004) tells us that channel concavity is less sensitive to grain size when there is a mixture of sizes available to the river.

Optional exercise: *Make a copy of* `tlim2.in` *and configure it to re-start from* `tlim2` *but with a higher uplift rate. Use the Matlab script* `cstrmproseries` *to plot fluvial profiles undergoing transient response. How do these compare with the detachment-limited model?*

7.3. **Hybrid Model: Combining Detachment and Transport.** Next, we'll look at a more complex situation with simultaneous erosion and sedimentation, and simultaneous detachment-limited and transport-limited behavior. In this case, we use a fluvial model in which erosion rate can be limited either by transport capacity or by detachment capacity, depending on their relative magnitudes:

$$(18) \qquad E_i = \begin{cases} \frac{Q_c - \sum_{j=1}^{N_i} Q_{sij}}{\Lambda_i} & \text{if } \frac{Q_c - \sum_{j=1}^{N_i} Q_{sij}}{\Lambda_i} < D_c \\ D_c & \text{otherwise} \end{cases}$$

Erosion and Deposition, Together at Last.

(1) *In the terminal window, navigate to the* `Hybrid` *folder and run:*
    `../child erodep1.in`

The 1m.y. run should take about 5 minutes (but of course you can peek at earlier time steps while the run is going, by reducing the number of frames in your movie).

(2) *In Matlab:* `figure(1), clf`

(3) `cmovie( 'erodep1', 21, 6e4, 6e4, 4000 );`

Here we have a block rising at 1mm/yr and an adjacent block subsiding at 0.25mm/yr. Uplift and subsidence shut down after 500ky. The subsiding block forms a large lake that gradually fills in with fan-deltas.

7.4. **Other Sediment-Flux-Dependent Fluvial Models.**

- "Under-capacity" models (detachment rate depends on degree to which sediment flux falls below transport capacity)
- Saltation-abrasion models (detachment rate driven by particle impacts, and limited by alluvial shielding of bed)
- Gasparini et al. (2007) explore behavior of these models with CHILD simulations

8. Multiple Grain Sizes

Interesting issues regarding grain size include:

- Bed armoring and its impact on transport rates
- Downstream fining
- Abrasion and lithologic controls

9. Exotica

- Stream meandering in the context of landscape evolution and valley stratigraphy (Clevis et al., 2006a,b)
- Vegetation (Collins et al., 2004; Istanbulluoglu and Bras, 2005)
- Alternate forms of mass wasting (Densmore et al., 1998; Lancaster et al., 2003; Istanbulluoglu et al., 2005)
- Knickpoints, hanging valleys, plunge pools (Flores-Cervantes et al., 2006; Crosby et al., 2007)
- Glaciation (Herman and Braun, 2006; Herman et al., 2007; Herman and Braun, 2008)

10. Forecasting or Speculation?

Some mathematical models in the physical sciences have such firm foundations that they can be relied upon to forecast the behavior of the natural world. For example, laws of motion of objects in a vacuum are absolutely reliable (as long as their speed is much less than that of light). The same can be said for numerical

solutions to these equations, provided the solution is reasonably accurate. For these kinds of model, the verb "to model" means to calculate with high reliability what would happen under a particular set of initial and boundary conditions.

At the other end of the spectrum, we have mathematical models that are essentially tentative hypotheses. Such models are often based on intuition about a physical system, and represent a sort of educated guess about the quantitative relationships between things. For example, when Ahnert (1976) presented his inverse-exponential equation for regolith generation from bedrock, he was essentially expressing a conceptual hypothesis in mathematical terms. For these models-as-hypotheses, the phrase "to model" means to perform a quantitative "what if" experiment, asking the question: what kinds of pattern would I see if my hypothesis were correct? Comparing the prediction with observations provides a test of the hypothesis.

One can find many models that fall between these extremes. There are models that are based on well-known physics, but which are forced to use approximations of unknown accuracy in order to solve the governing equations. For example, climate models typically use simple parameterization schemes to represent convective mass and energy transport. Then too there are models that combine basic physical principles with elements of intuition, empiricism, and approximation. Arguably, many sediment-transport laws fall into this category: they are based on firm mechanical foundations (the force balance on a sediment grain) but also rely on strong approximations of factors like grain geometry, local flow velocity, and so on.

By now, it should be obvious that landscape evolution models also fall somewhere between the end-member cases of "model as truth" and "model as speculative hypothesis." As we saw on Day 1, there is a varying degree of experimental and observational support for the individual transport, weathering and erosion laws that go into a typical landscape model. In that sense, then, these models amount to more than just speculation. But equally there is still an element of speculation behind many of the process laws used in landscape models. Also, the process laws and algorithms represent a significant amount of upscaling in space and (especially) time. For example, the use of a steady precipitation rate as a proxy for the natural sequence of flows in a river channel represents a major approximation. For these reasons, I believe that three of the most important frontiers in landscape evolution research are (1) continuing to test individual process laws in the field and lab, (2) testing whole-landscape models using natural experiments, and (3) using mathematics, computation and experiments to study how the rates of various processes scale upward in time and space, and how these can be effectively parameterized.

11. Ten Commandments of Landscape Evolution Modeling

(1) Thou shalt not use a model without understanding the ingredients therein.

(2) Be thou ever mindful of uncertainty.
(3) Thou shalt use thy model to develop insight.
(4) Thou shalt take delight when thy model surprises thee.
(5) Thou shalt kick thy model hard, that it may notice thee (an injunction borrowed gratefully from the 10 Climate Modeling Commandments).
(6) Thou shalt diagnose the reasons for thy model's behavior.
(7) Thou shalt conduct sensitivity experiments and "play around."
(8) Thou shalt use thy model to discover the necessary and sufficient conditions needed to explain thy target problem.
(9) If thou darest use a model to calculate what happened in your field area in the past, thou shalt find a way to test and calibrate it first.
(10) If thou darest to predict future erosion, thou shalt heed the previous commandment ten times over (but thou mightest point out to skeptics that a process-based prediction is usually better than one based on pure guesswork, provided that commandment #2 is obeyed).

References

Ahnert, F., 1976: Brief description of a comprehensive three-dimensional process-response model of landform development. *Zeitschrift für Geomorfologie, Supplementband*, **25**, 29–49.

Attal, M., G. E. Tucker, A. C. Whittaker, P. A. Cowie, and G. P. Roberts, 2008: Modeling fluvial incision and transient landscape evolution: Influence of dynamic channel adjustment. *Journal of Geophysical Research*, **113**, F03013.

Beaumont, C., P. Fullsack, and J. Hamilton, 1992: Erosional control of active compressional orogens. *Thrust tectonics*, **99**, 1–18.

Braun, J. and M. Sambridge, 1997: Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization. *Basin Research*, **9**, 27–52.

Carretier, S. and F. Lucazeau, 2005: How does alluvial sedimentation at range fronts modify the erosional dynamics of mountain catchments? *Basin Research*, **17(3)**, 361–381.

Chase, C. G., 1992: Fluvial landsculpting and the fractal dimension of topography. *Geomorphology*, **5**, 39–57.

Clevis, Q., G. E. Tucker, S. T. Lancaster, A. Desitter, N. Gasparini, and G. Lock, 2006a: A simple algorithm for the mapping of TIN data onto a static grid: Applied to the stratigraphic simulation of river meander deposits. *COMPUTERS & GEOSCIENCES*, **32(6)**, 749–766.

Clevis, Q., G. E. Tucker, G. Lock, S. T. Lancaster, N. Gasparini, A. Desitter, and R. L. Brass, 2006b: Geoarchaeological simulation of meandering river deposits and

settlement distributions: A three-dimensional approach. *GEOARCHAEOLOGY-AN INTERNATIONAL JOURNAL*, **21(8)**, 843–874.

Collins, D., R. Bras, and G. Tucker, 2004: Modeling the effects of vegetation-erosion coupling on landscape evolution. *JOURNAL OF GEOPHYSICAL RESEARCH-EARTH SURFACE*, **109(F3)**.

Coulthard, T., M. Kirkby, and M. Macklin, 1996: A cellular automaton landscape evolution model. In *Proceedings of the First International Conference on GeoComputation*, vol. 1, pp. 248–81.

Crave, A. and P. Davy, 2001: A stochastic 'precipiton' model for simulating erosion/sedimentation dynamics. *Computers and Geosciences*, **27**, 815–827.

Crosby, B. T., K. X. Whipple, N. M. Gasparini, and C. W. Wobus, 2007: Formation of fluvial hanging valleys: Theory and simulation. *JOURNAL OF GEOPHYSICAL RESEARCH-EARTH SURFACE*, **112(F3)**.

Densmore, A. L., M. A. Ellis, and R. S. Anderson, 1998: Landsliding and the evolution of normal-fault-bounded mountains. *Journal of Geophysical Research*, **103**, 15203–15219.

Fagherazzi, S., A. Howard, and P. Wiberg, 2002: An implicit finite difference method for drainage basin evolution. *Water Resources Research*, **38(7)**, 21.

Flores-Cervantes, H., E. Istanbulluoglu, and R. L. Bras, 2006: Development of gullies on the landscape: A model of headcut retreat resulting from plunge pool erosion. *Journal of Geophysical Research*, **111**, F01010.

Garcia-Castellanos, D., 2002: Interplay between lithospheric flexure and river transport in foreland basins. *Basin Research*, **14(2)**, 89–104.

Gasparini, N., G. Tucker, and R. Bras, 1999: Downstream fining through selective particle sorting in an equilibrium drainage network. *Geology*, **27(12)**, 1079.

—, 2004: Network-scale dynamics of grain-size sorting: Implications for downstream fining, stream-profile concavity, and drainage basin morphology. *EARTH SURFACE PROCESSES AND LANDFORMS*, **29(4)**, 401–421.

Gasparini, N. M., K. X. Whipple, and R. L. Bras, 2007: Predictions of steady state and transient landscape morphology using sediment-flux-dependent river incision models. *JOURNAL OF GEOPHYSICAL RESEARCH-EARTH SURFACE*, **112(F3)**.

Gilbert, G., 1877: Report on the geology of the Henry Mountains: US Geog. and Geol. *Survey, Rocky Mtn. Region*, **160**.

Herman, F. and J. Braun, 2006: Fluvial response to horizontal shortening and glaciations: a study in the Southern Alps of New Zealand. *Journal of Geophysical Research-Earth Surface*, **111(F1)**, F01008.

—, 2008: Evolution of the glacial landscape of the Southern Alps of New Zealand: Insights from a glacial erosion model. *Journal of Geophysical Research-Earth Surface*, **113(F2)**, F02009.

Herman, F., J. Braun, and W. Dunlap, 2007: Tectonomorphic scenarios in the Southern Alps of New Zealand. *Journal of Geophysical Research-Solid Earth*, **112(B4)**, B04201.

Howard, A. D., 1994: A detachment-limited model of drainage basin evolution. *Water Resources Research*, **30(7)**, 2261–2285.

Huang, X. and J. Niemann, 2006: An evaluation of the geomorphically effective event for fluvial processes over long periods. *Journal of Geophysical Research*, **111(F3)**, F03015.

Istanbulluoglu, E. and R. L. Bras, 2005: Vegetation-modulated landscape evolution: Effects of vegetation on landscape processes, drainage density, and topography. *Journal of Geophysical Research*, **110**, F02012.

Istanbulluoglu, E., R. L. Bras, H. Flores-Cervantes, and G. E. Tucker, 2005: Implications of bank failures and fluvial erosion for gully development: Field observations and modeling. *Journal of Geophysical Research*, **110**, F01014.

Lancaster, S., S. Hayes, and G. Grant, 2003: Effects of wood on debris flow runout in small mountain watersheds. *Water Resources Research*, **39(6)**, 1168.

Mitas, L. and H. Mitasova, 1998: Distributed soil erosion simulation for effective erosion prevention. *Water Resources Research*, **34(3)**, 505–516.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 2007: *Numerical recipes: the art of scientific computing*. Cambridge Univ Pr.

Schoorl, J., A. Veldkamp, and J. Bouma, 2002: Modeling water and soil redistribution in a dynamic landscape context. *Soil Science Society of America Journal*, **66(5)**, 1610.

Snyder, N. P., K. X. Whipple, G. E. Tucker, and D. M. Merritts, 2003: Importance of a stochastic distribution of floods and erosion thresholds in the bedrock river incision problem. *Journal of Geophysical Research*, **108**, 2117.

Sólyom, P. and G. Tucker, 2004: Effect of limited storm duration on landscape evolution, drainage basin geometry, and hydrograph shapes. *Journal of Geophysical Research*, **109**, 13.

Tucker, G., 2004: Drainage basin sensitivity to tectonic and climatic forcing: Implications of a stochastic model for the role of entrainment and erosion thresholds. *EARTH SURFACE PROCESSES AND LANDFORMS*, **29(2)**, 185–205.

Tucker, G. E. and R. L. Bras, 2000: A stochastic approach to modeling the role of rainfall variability in drainage basin evolution. *Water Resources Research*, **36(7)**, 1953–1964.

Tucker, G. E. and G. R. Hancock, 2010: Modelling landscape evolution. *Earth Surface Processes and Landforms*, **46**, 28–50.

Tucker, G. E., S. T. Lancaster, N. M. Gasparini, R. L. Bras, and S. M. Rybarczyk, 2001: An object-oriented framework for hydrologic and geomorphic modeling using triangular irregular networks. *Computers and Geosciences*, **27**, 959–973.

Tucker, G. E. and R. L. Slingerland, 1994: Erosional dynamics, flexural isostasy, and long-lived escarpments: A numerical modeling study. *Journal of Geophysical Research*, **99**, 12,229–12,243.

Whipple, K. X. and G. E. Tucker, 1999: Dynamics of the stream-power river incision model: Implications for height limits of mountain ranges, landscape response timescales, and research needs. *Journal of Geophysical Research*, **104**, 17661–17674.

Willgoose, G., R. L. Bras, and I. Rodriguez-Iturbe, 1991: A coupled channel network growth and hillslope evolution model, 1, theory. *Water Resources Research*, **27(7)**, 1671–1684.

Wobus, C., G. Tucker, and R. Anderson, 2006: Self-formed bedrock channels. *Geophys. Res. Lett*, **33**, 1–6.