

Landlab: A Python library for building and coupling 2D numerical models

University of Colorado Boulder

CSDMS
COMMUNITY SURFACE DYNAMICS MODELING SYSTEM



Gregory E. Tucker^{1,4}, Daniel E.J. Hobley¹, Nicole M. Gasparini², Jordan M. Adams²,
Sai Nudurupati³, Erkan Istanbulluoglu³, Eric Hutton⁴, and Jenny Knuth¹

Tulane University

UNIVERSITY OF
WASHINGTON

1 - Cooperative Institute for Research in Environmental Sciences (CIRES)
and Department of Geological Sciences, University of Colorado at Boulder

2 - Department of Earth and Environmental Sciences, Tulane University

3 - Department of Civil and Environmental Engineering, University of Washington, Seattle

4 - Community Surface Dynamics Modeling System (CSDMS), University of Colorado at Boulder

ABSTRACT

Landlab is a Python-language library that speeds the building and coupling of 2D numerical models, with a primary focus on models of earth-surface processes. Landlab's capabilities include: (1) create and configure a grid with one or a few lines of code; (2) choose from a variety of grid types, either structured and unstructured; (3) connect data arrays directly to the grid, where they can be accessed and shared; (4) assemble integrated models from reusable components, each of which models a particular process; (4) perform input and output using standardized data formats; and (5) design continuous-time stochastic cellular automaton models by specifying cell states and a set of transition rules. More information about Landlab is available at <http://landlab.github.io>.

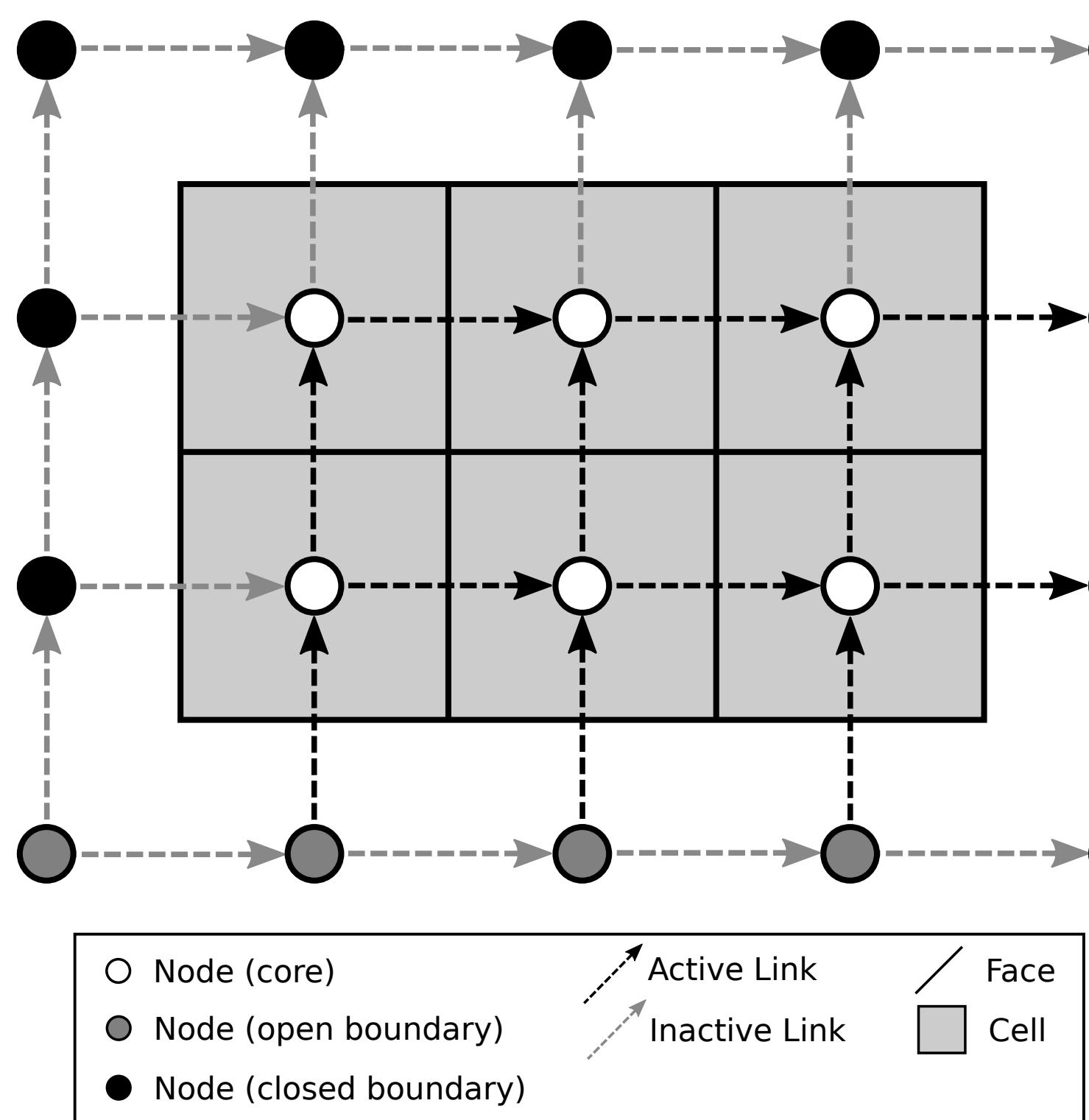


What is Landlab?

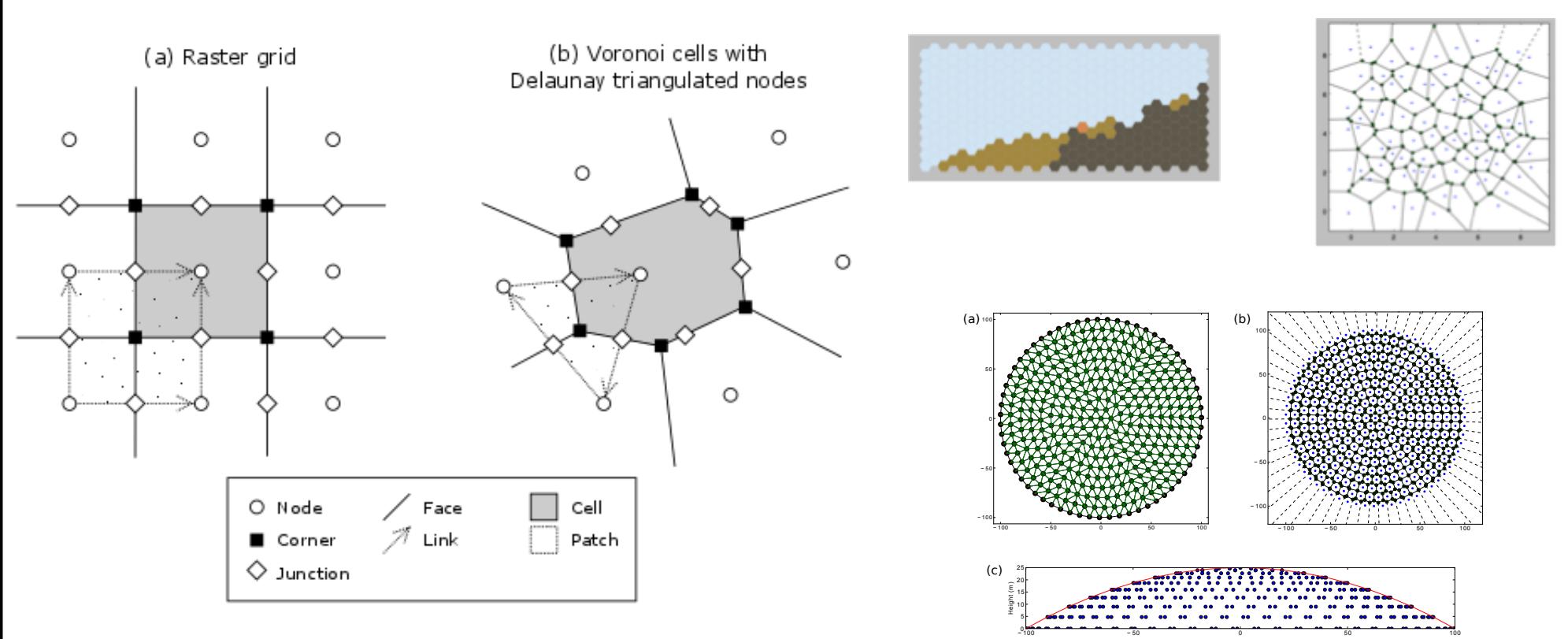
- An open-source, Python-language library that helps geoscience researchers efficiently develop 2D grid-based numerical models
- A set of pre-built model components, each of which models a particular landscape process (see examples below)
- A framework for combining components into multi-process models
- Learn more at <http://landlab.github.io>

GRIDS

Grids are built from primitives such as nodes, links, and cells

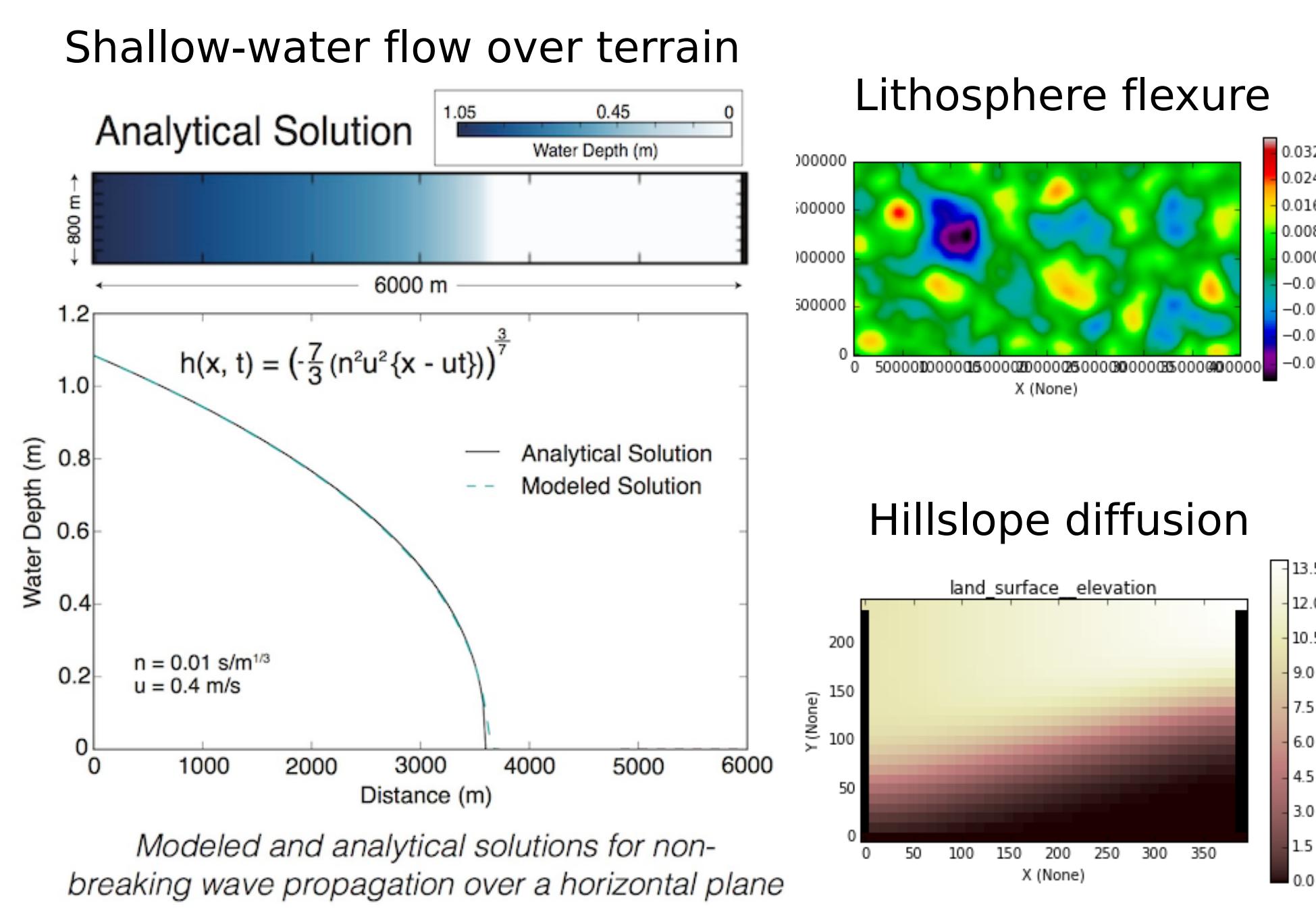


Different grid types

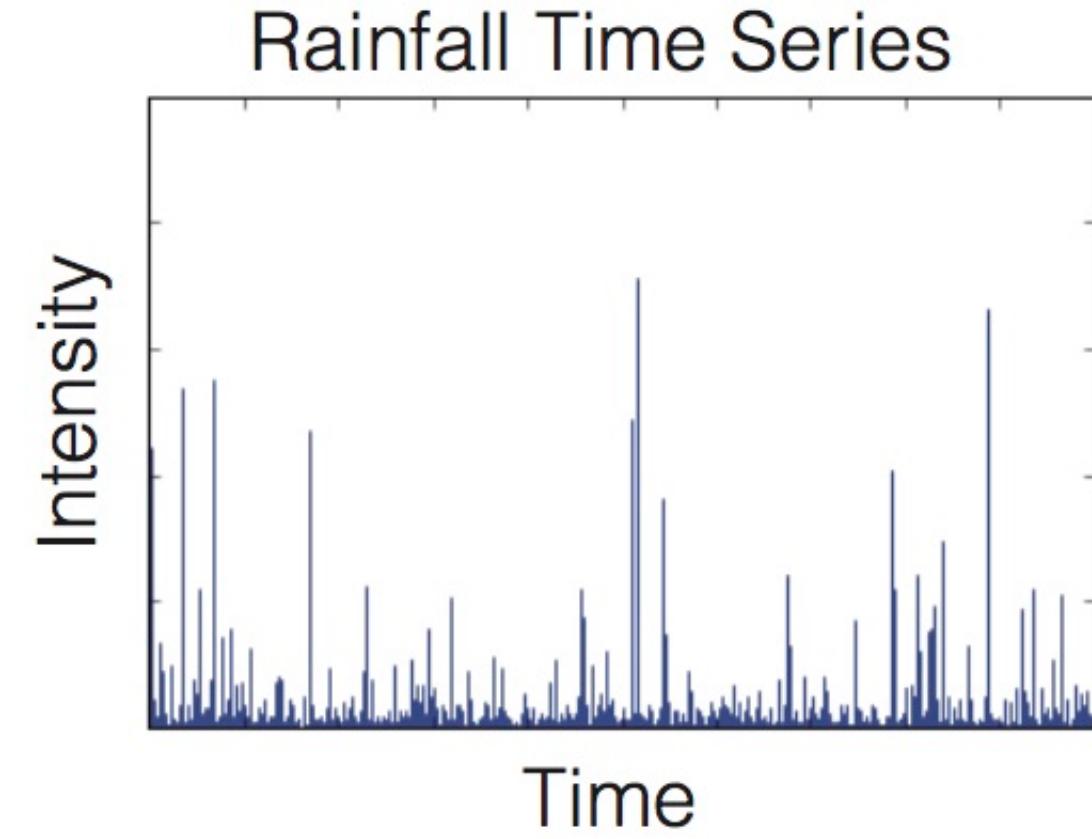


COMPONENTS

- Standard design
- Sharing data through grid object
- Coupling with driver script

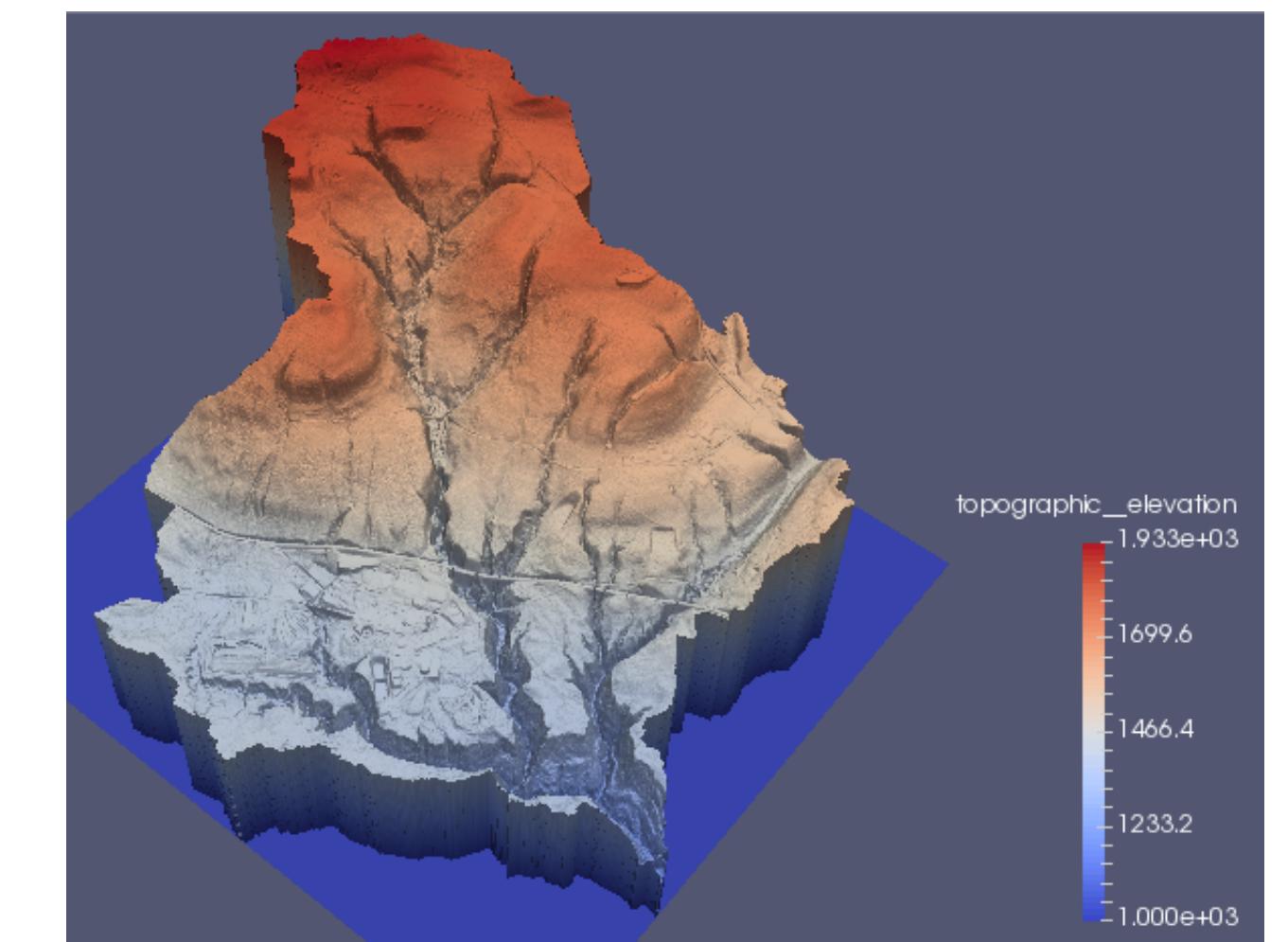


Stochastic rainfall generation



UTILITIES

- Import ESRI Arc AsciiGrid format digital elevation data
- Read and write netCDF files
- Read model parameters from formatted text files



SCRIPTING

Example: a nine-line diffusion model

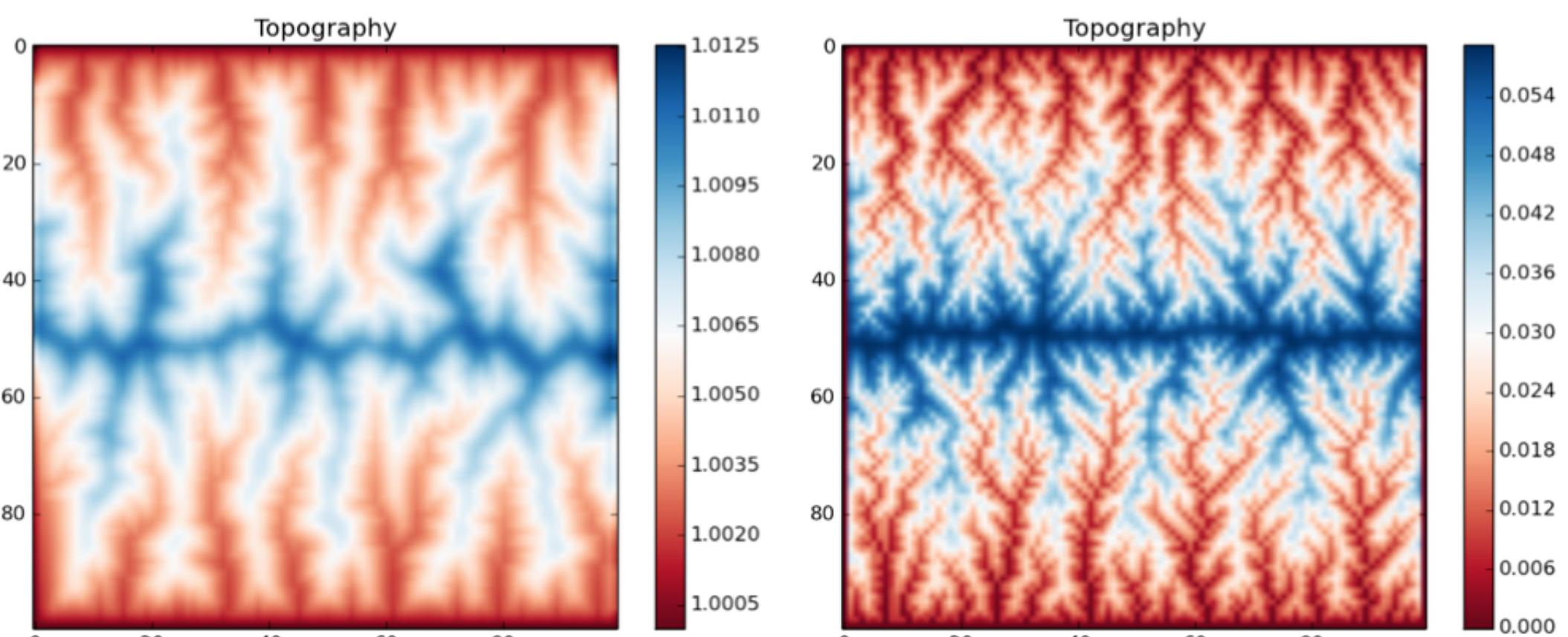
```
mg = landlab.RasterModelGrid(numrows, numcols, dx)
z = mg.add_zeros('node', 'land_surface_elevation')
core_nodes = mg.core_nodes()

for i in range(0, num_time_steps): # main loop
    g = mg.calculate_gradients_at_active_links(z) # slope
    qs = -kdg # sediment flux
    dgsds = mg.calculate_flux_divergence_at_nodes(qs)
    dzdt = uplift_rate - dgsds
    z[core_nodes] += dzdt[core_nodes] * dt

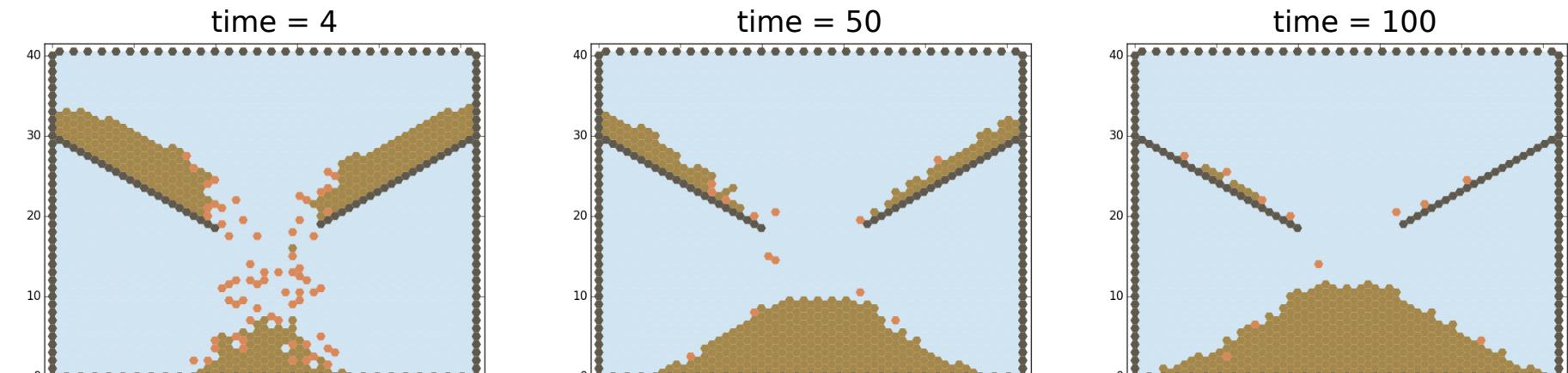
    # Simulated topography with uplift and diffusion
```

EXAMPLES

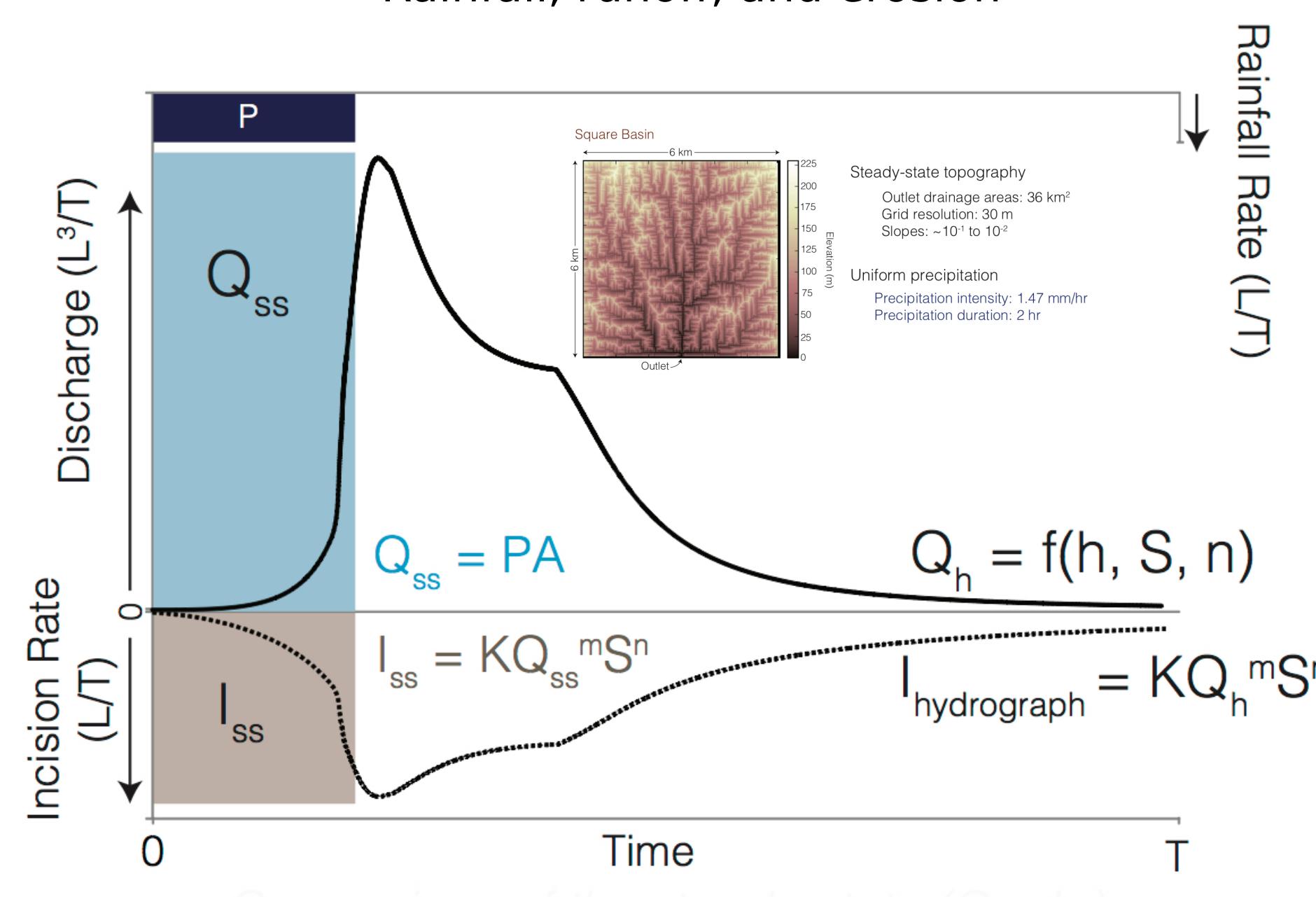
Landform Evolution



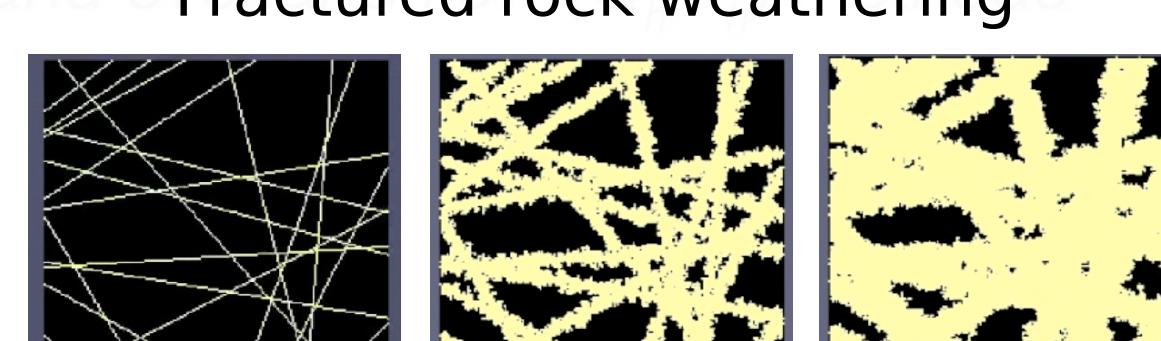
Granular mechanics (lattice-grain model)



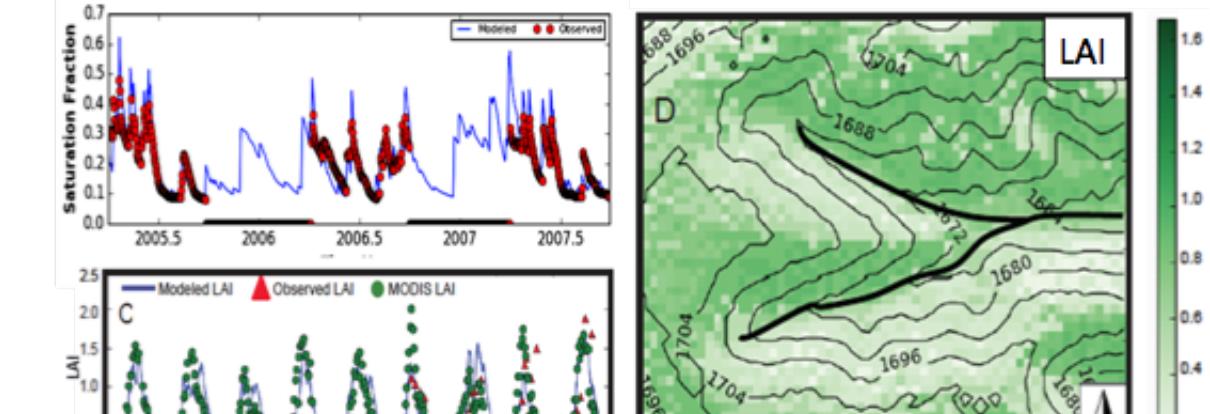
Rainfall, runoff, and erosion



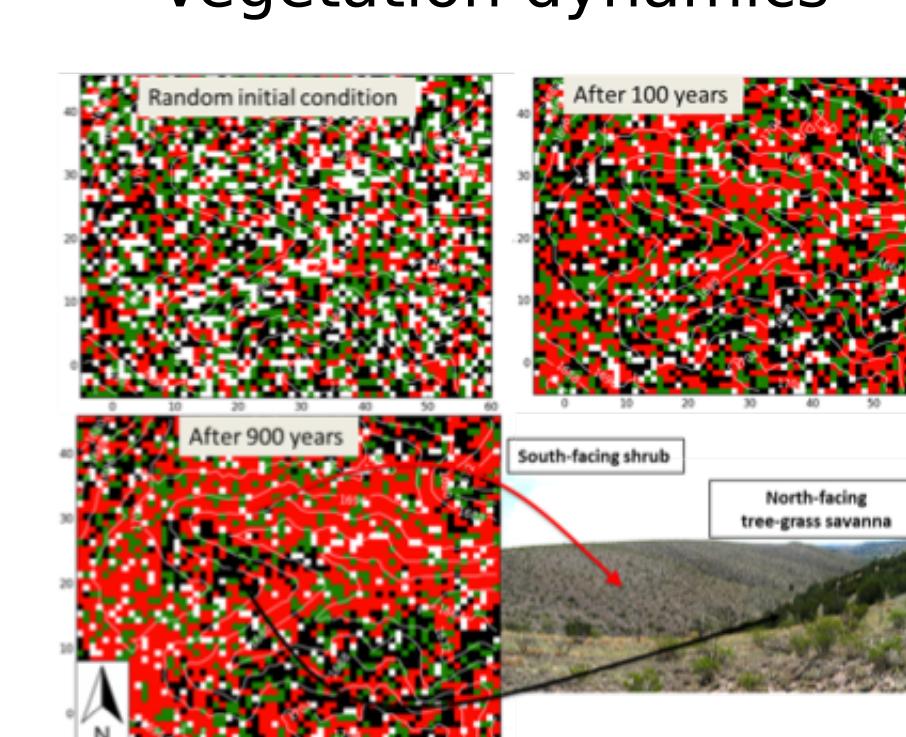
Fractured rock weathering



Soil moisture and leaf-area index (LAI)



Vegetation dynamics



Acknowledgments:
Development of Landlab is supported by
NSF (ACI-1450409 and ACI-1147454)