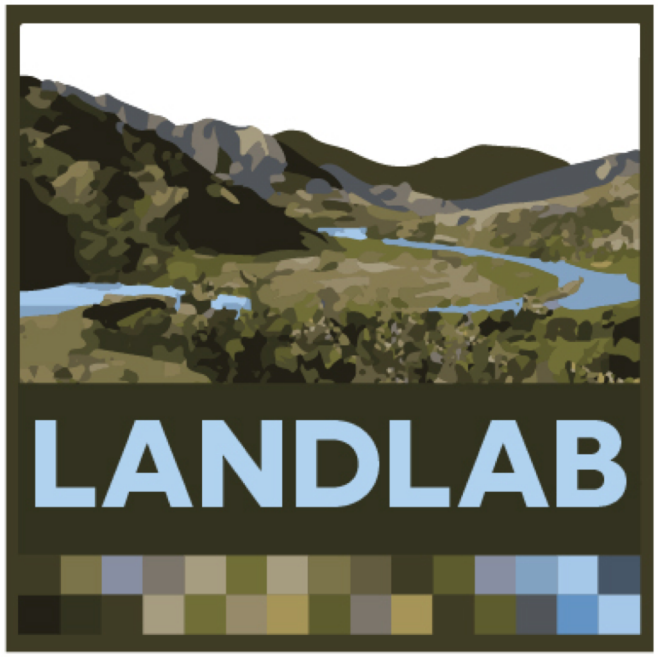


CSDMS
COMMUNITY SURFACE DYNAMICS MODELING SYSTEM

Modeling earth-surface dynamics with Landlab

Greg Tucker
Eric Hutton
Katy Barnhart
Jennifer Knuth
Jordan Adams
Margaux Mouchené

University of Colorado Boulder

Nicole Gasparini
Nathan Lyons

Tulane University

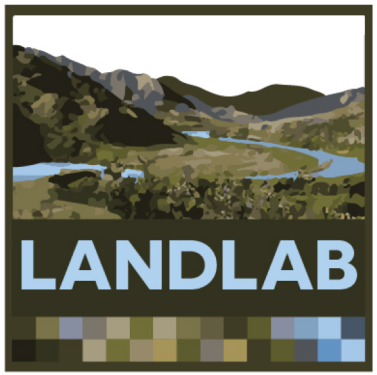
Erkan Istanbulluoglu
Sai Nudurupati
Christina Bandaragoda
Ronda Strauch
Claire Beveridge

UNIVERSITY of WASHINGTON

Dan Hobley

CARDIFF UNIVERSITY

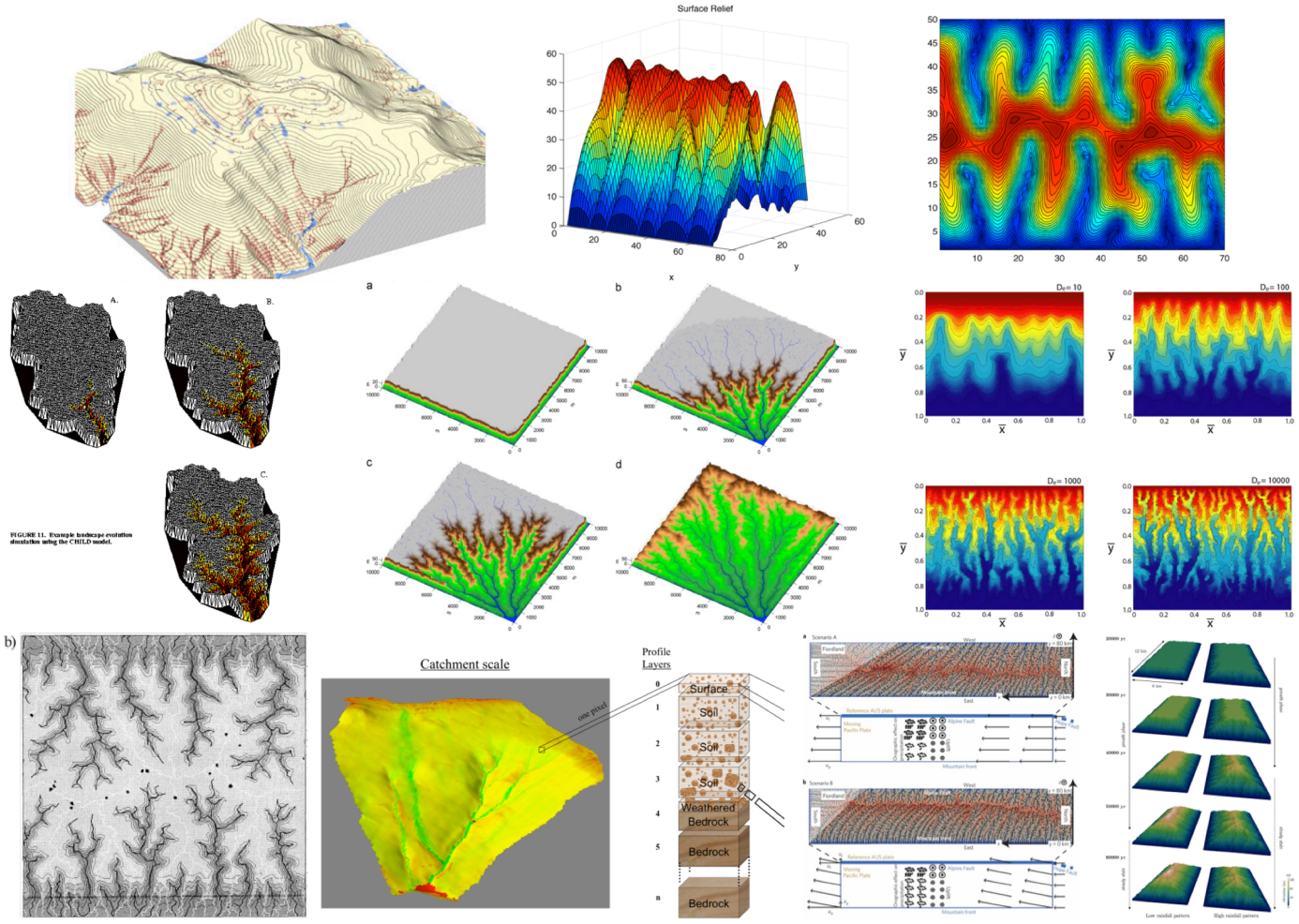
Overview



- Introduction to Landlab
- Running Landlab on Hydroshare → hydroshare.org

Motivation for Landlab

Many models exist:



SIBERIA (Willgoose and Riley, 1998), J. Prancevic (unpubl.), CHILD (Tucker et al., 2001), SIGNUM (Refice et al. 2012), Simpson & Schlunegger 2003, Braun & Sambridge 1997, mARM4D (Cohen et al., 2013), Castelltort et al. 2012, Tellus (CSIRO)

Motivation for Landlab

Many models exist:

- Built for specific goals → ~~low adaptability~~, ~~no/poor documentation and maintenance~~
- Costly (time + money) → ~~low efficiency and accessibility~~
- Closed source → ~~high cost, poor code assessment and reproducibility~~
- No standardization → ~~no interoperability~~
- Common elements → ~~duplicates~~

A single code, flexible, documented and maintained, open-source, quick to learn and write, with a standard interface

= LANDLAB

What is Landlab?

- an **open-source, collaborative, Python-based library** to build numerical landscape models
- Provides:
  - Grid structures and data storage
  - Robust, reusable components describing processes
  - Tools for data I/O, treatment and visualization
  - Tools to facilitate further development
- And also:
  - Documentation
  - Teaching tools
  - Compatibility with CSDMS WMT

What is Landlab?

Modular nature of Landlab makes it ideal for:

- A wide range of applications
- Coupling to third-party model
- Sharing, comparing, reproducing
- Classroom use
- Model intercomparison and epistemic uncertainty and sensitivity analyses

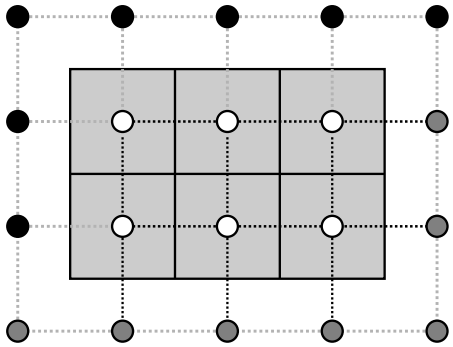
→ go to **"Landlab and Dakota" clinic** tomorrow!

The Landlab library

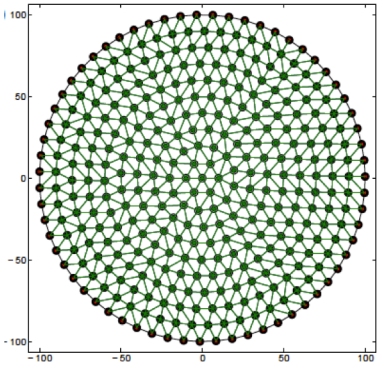
1. Landlab grid

- Create a structured or unstructured grid in a single line of code
- Common elements: nodes, cells…
- Attach data to grid elements = fields
- Built in numerical functions, e.g.: gradient, divergence

RASTER

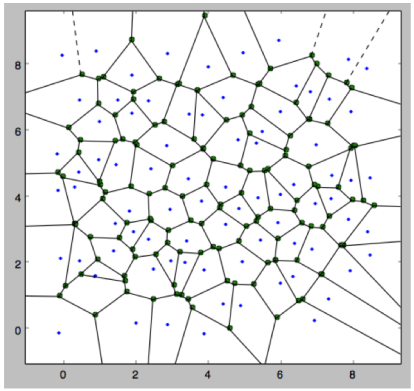


RADIAL

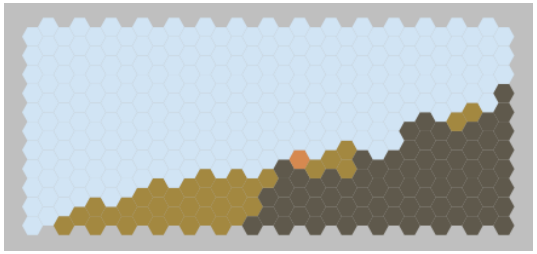


+ coming soon:
Network grid

VORONOI / DELAUNAY

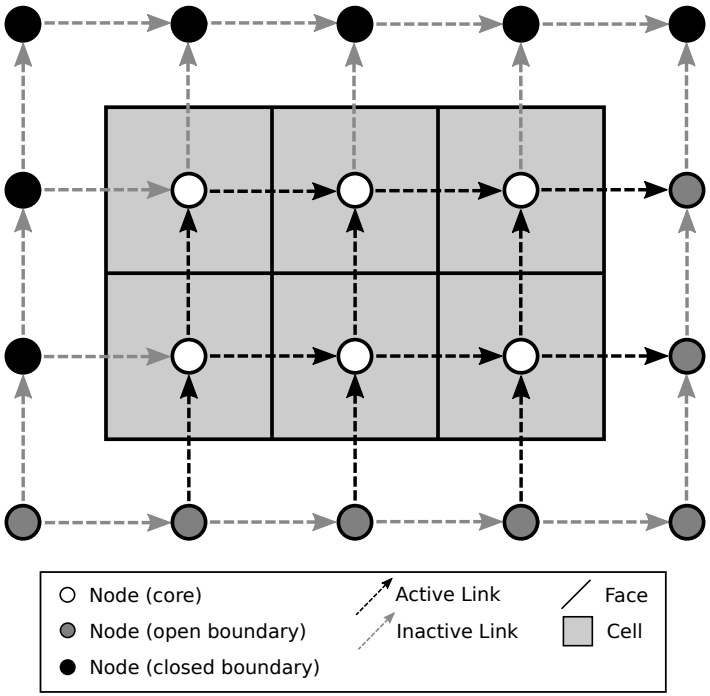


HEXAGONAL



Example: creating a grid

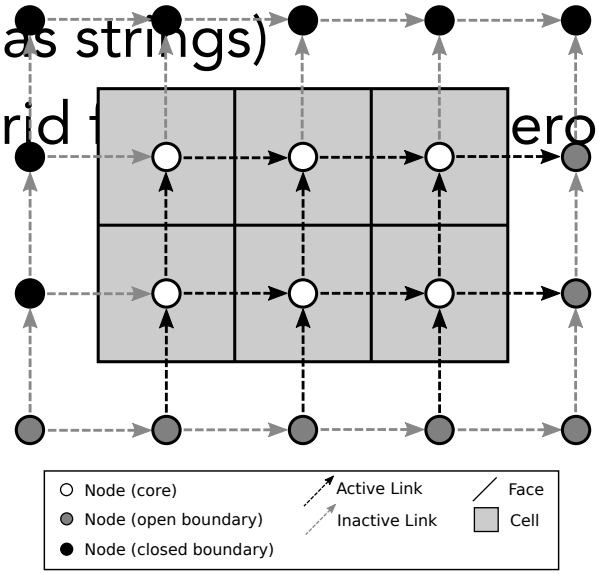
```
>>> from landlab import RasterModelGrid
>>> rg = RasterModelGrid((4, 5), 10.0)
>>> rg.number_of_nodes
20
```



Fields:
attaching data to the grid

- A **field** is a NumPy array containing data that are associated with a particular type of grid element (typically nodes or links)
- Fields are 1D
- Fields are number-of-element long:

  Values correspond to the element with the same ID. Example: value 6 of a node field belongs to node #6.
- Fields have names (as strings)
- Create fields with grid f eros, add_ones, or add_empty



2. Landlab components
and coupling framework for components

- A **component** models a **single process** (e.g., flow routing across terrain, ET, vegetation dynamics)
- Components have a **standard interface** and can be combined by writing a **short Python script (driver)**
- **Initialized** using input grid and parameters
- **Update** relevant **fields** on the grid

```
>>> from landlab.components import MyComponent
>>> component_1 = MyComponent(rg, param_1=1.0, …)
>>> component_1.run_one_step(dt=100)
```

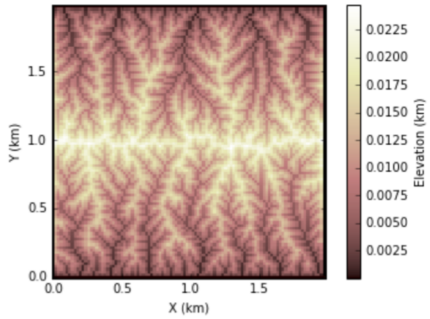
List of components

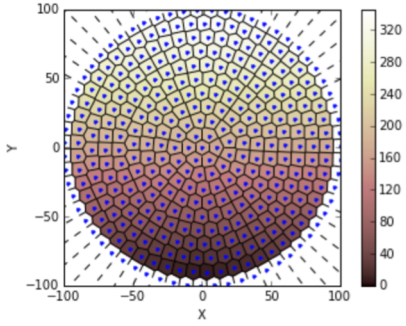
- **Hillslope diffusion**:
  - Linear
  - implicit nonlinear (Perron)
  - depth dependent (Johnstone and Hilley, 2014)
  - Transport-length non-local diffusion (Davy and Lague, 2009)
  - non-linear (Ganti et al. 2013)
- **Fluvial geomorphology:**
  - Stream-power erosion FastScape-style (+options)
  - "Tools and cover" effects
  - Detachment-limited
  - Fluvial erosion Davy and Lague (2009) style
  - Stream Power with Alluvium Conservation and Entrainment
- **Flow routing:**
  - Flow Direction (D4, D8, MFD, Dinf)
  - Flow Accumulation
  - Depression Finder And Router
  - Potentiality Flow Router
  - Sink Filler
- **Shallow water hydrodynamics**
  - Overland Flow (de Alameida, Bates)
- **Land surface hydrology**
  - Solar radiation
  - Potential Evapotranspiration
  - Soil Moisture
  - Soil Infiltration (GreenAmpt)
- **Landslides**
- **Vegetation**
- **Precipitation**
- **Weathering**
- **Fire**
- **Tectonics**
  - Flexure (uniform or variable rigidity)
  - Lithospheric deflection
  - Normal Fault
- **Terrain Analysis**
  - Steepness and concavity indices
  - Chi-index analysis

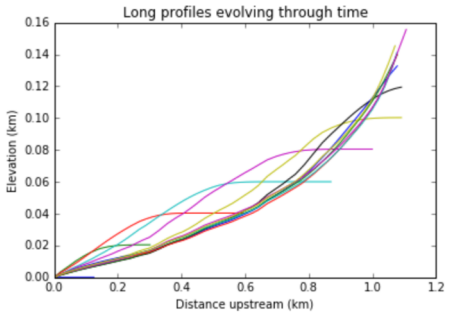
3. Data I/O, processing and visualization

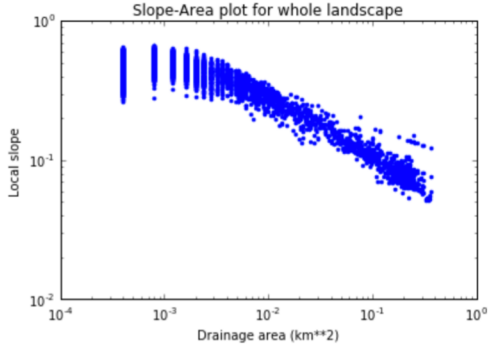
- Read model parameters from a formatted text file
- Read in digital terrain data:

  e.g., ESRI ASCII DEMs ➔ Landlab grid
- Write gridded output to files (netCDF format)
- Plot data using Matplotlib graphics library
- Analytical tools: drainage plot, channel profile, channel steepness, etc.





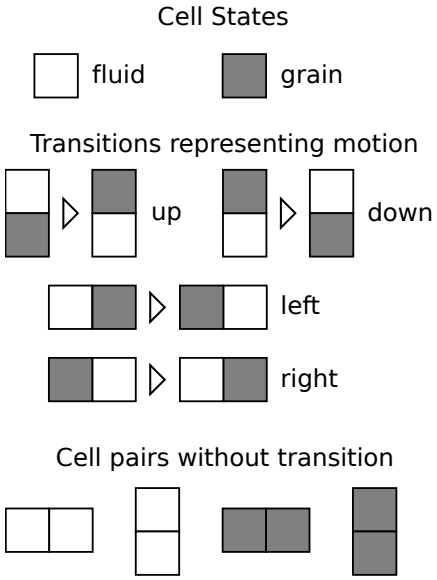


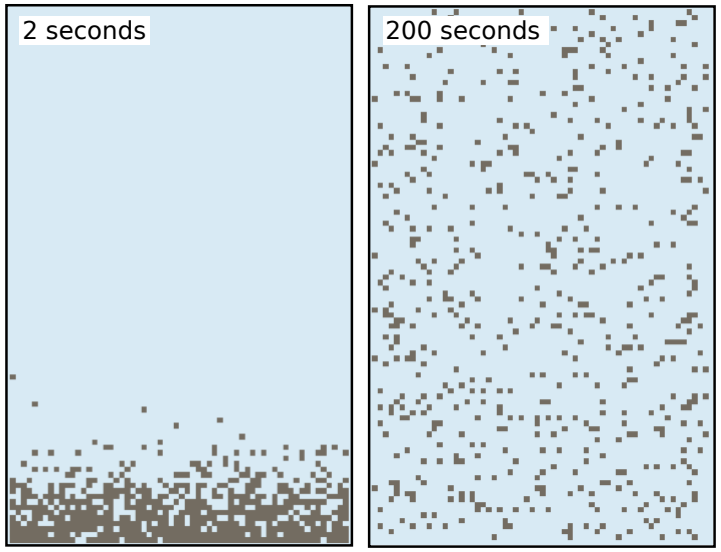


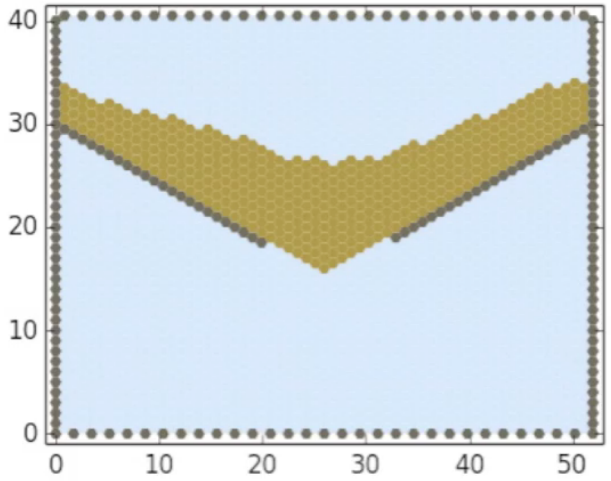
Cellular-automaton modeling

Support for cellular-automaton modeling

CellLab-CTS: Continuous-time stochastic CA model implementation







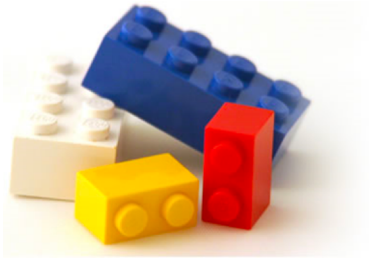
(Tucker et al., 2016 Geoscientific Model Development)

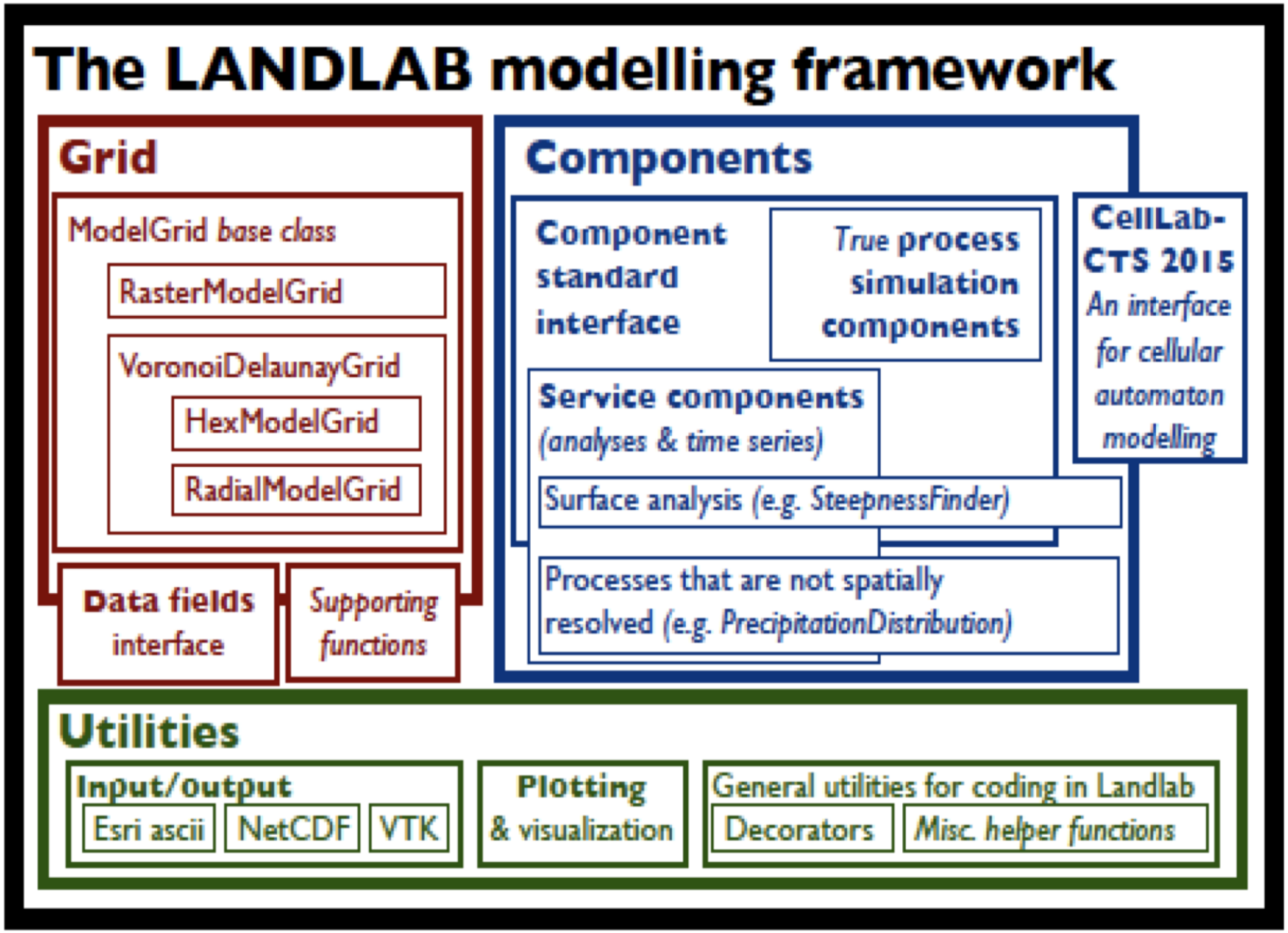
Coming soon!

- Network grid
- Species evolver
- Clast-tracker
- Layered lithology
- …your own component?

Summary

- Landlab is a Python-based, flexible, quick to learn, open-source library
- A number of existing process modules is available
- Users are encouraged to build their own!

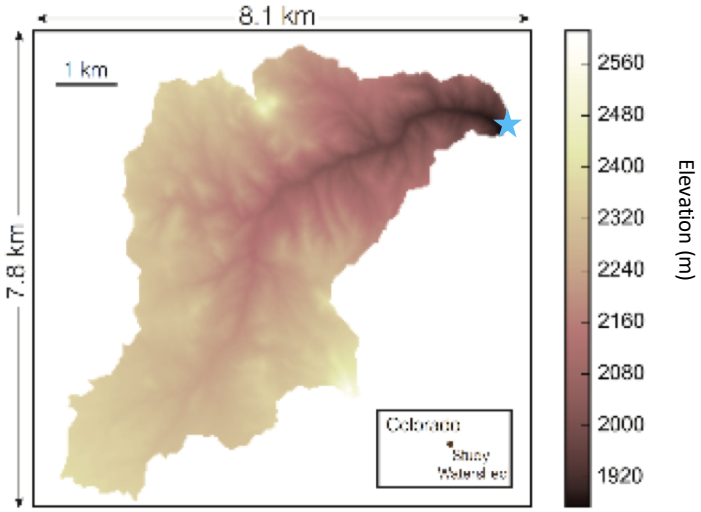


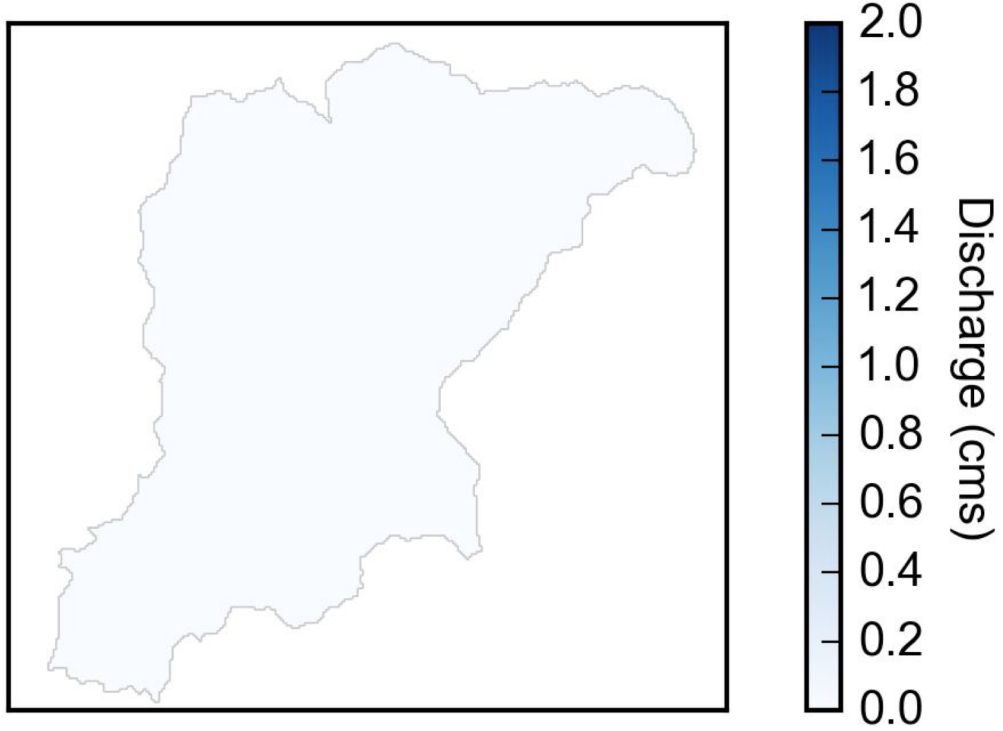

The LANDLAB modelling framework
Grid
ModelGrid base class
RasterModelGrid
VoronoiDelaunayGrid
HexModelGrid
RadialModelGrid
Data fields
interface
Supporting
functions
Components
Component
standard
interface
True process
simulation
components
CellLab-
CTS 2015
An interface
for cellular
automaton
modelling
Service components
(analyses & time series)
Surface analysis (e.g. SteepnessFinder)
Processes that are not spatially
resolved (e.g. PrecipitationDistribution)
Utilities
Input/output
Esri ascii
NetCDF
VTK
Plotting
& visualization
General utilities for coding in Landlab
Decorators
Misc. helper functions

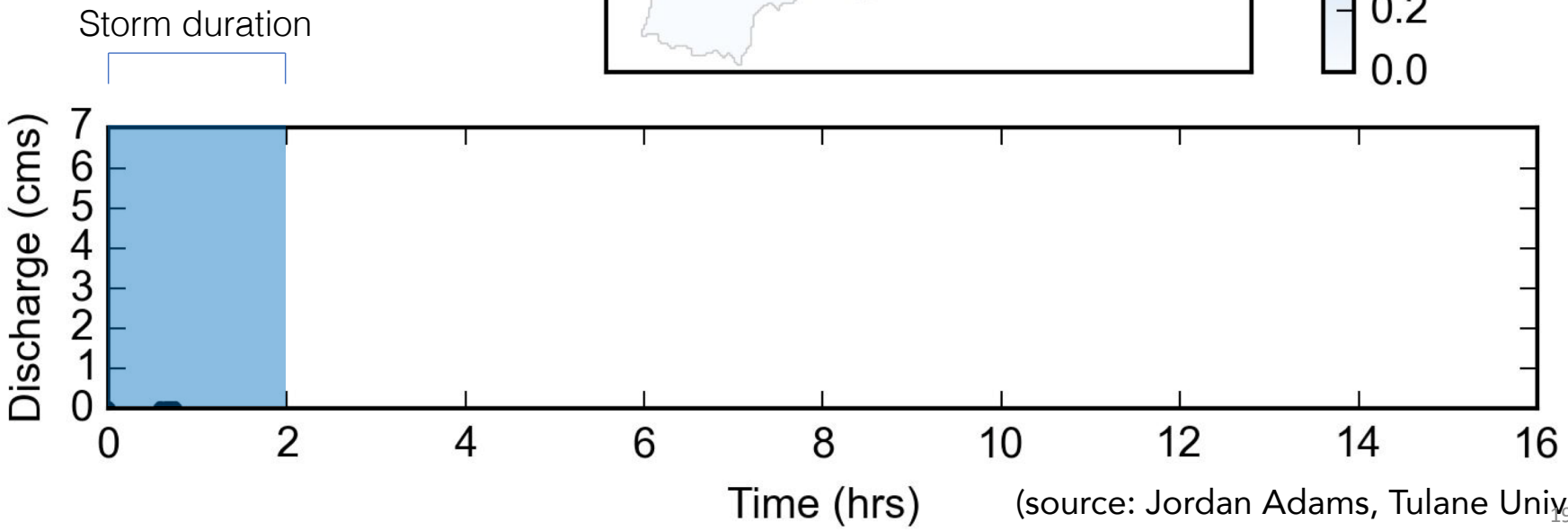
Examples

Overland Flow – hydrograph routing

Spring Creek, CO

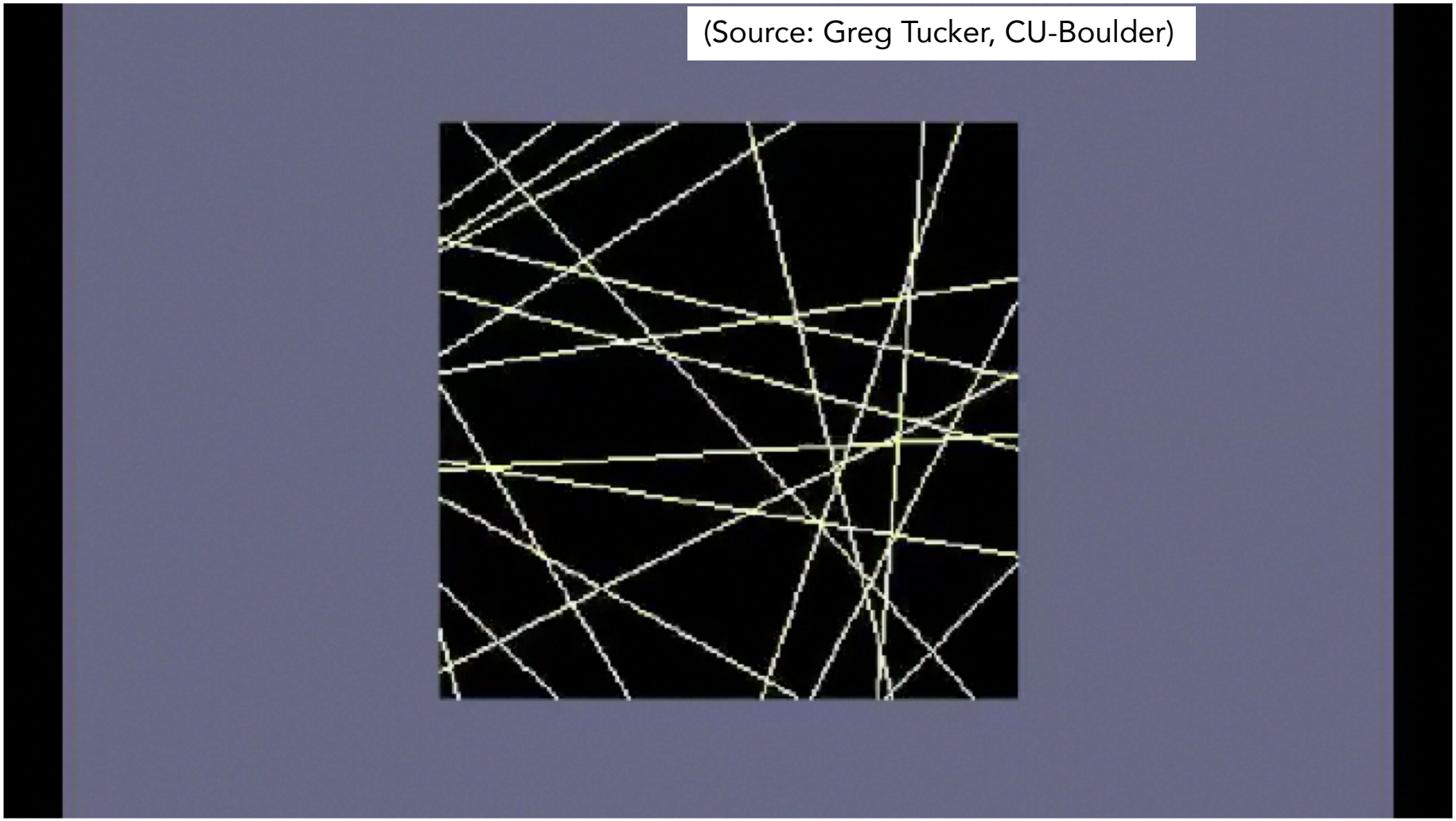




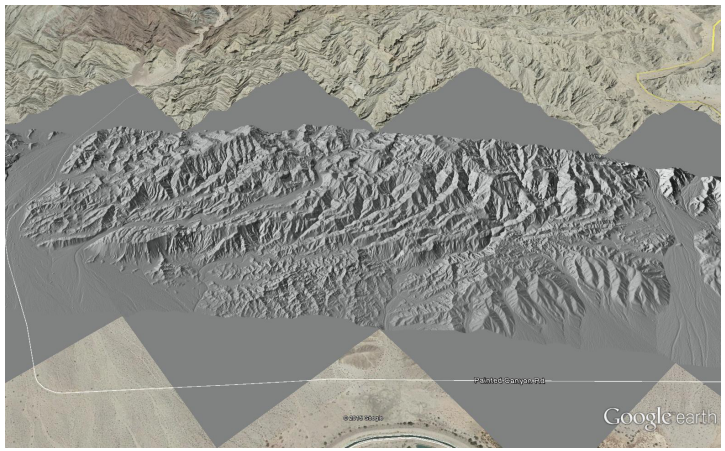


(source: Jordan Adams, Tulane University)

Cellular automaton model of weathering along fractures

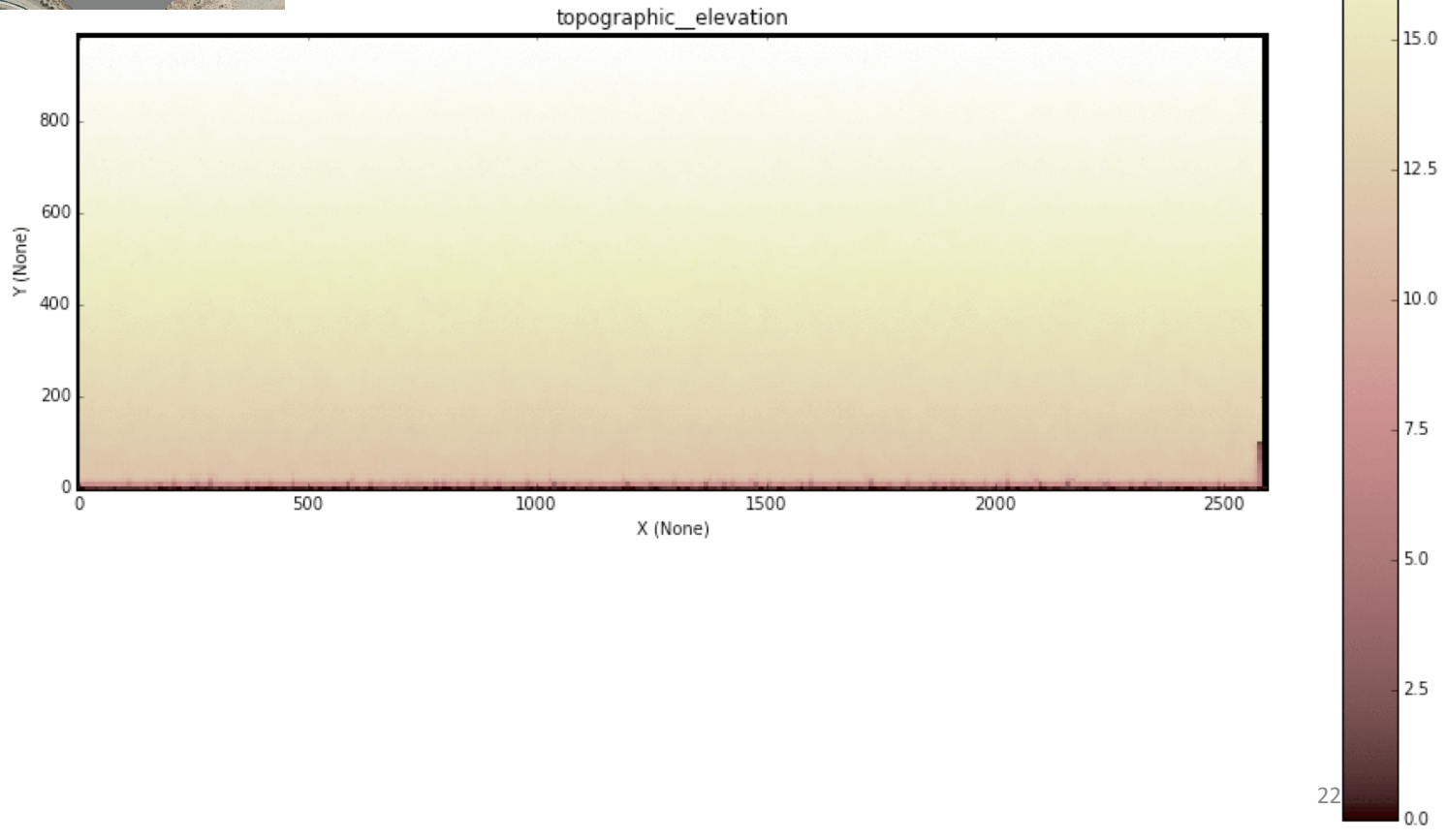


Shear on strike-slip faults

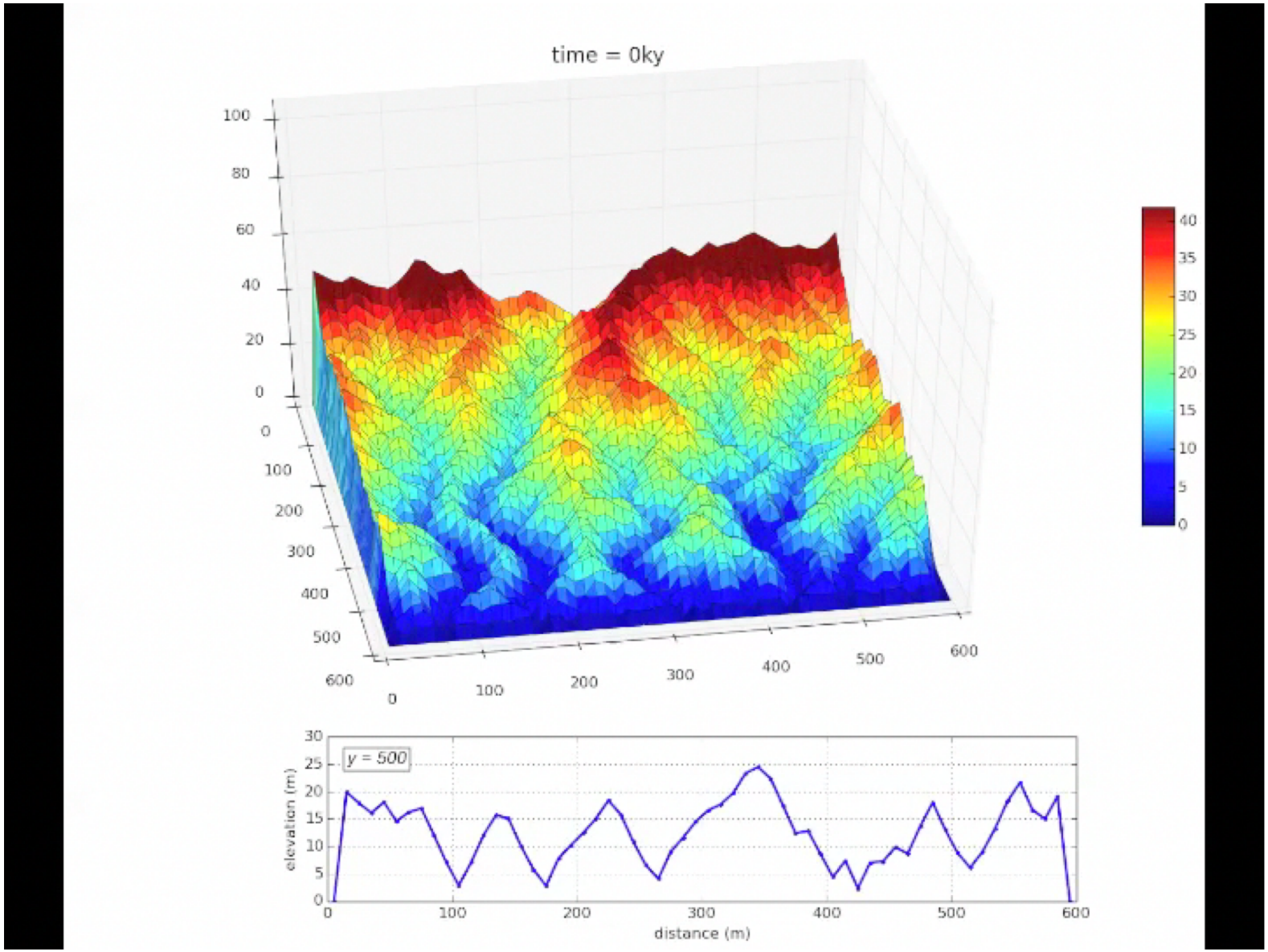


SAN ANDREAS FAULT, MECCA HILLS, CA

(Source: Harrison Gray, CU-Boulder)

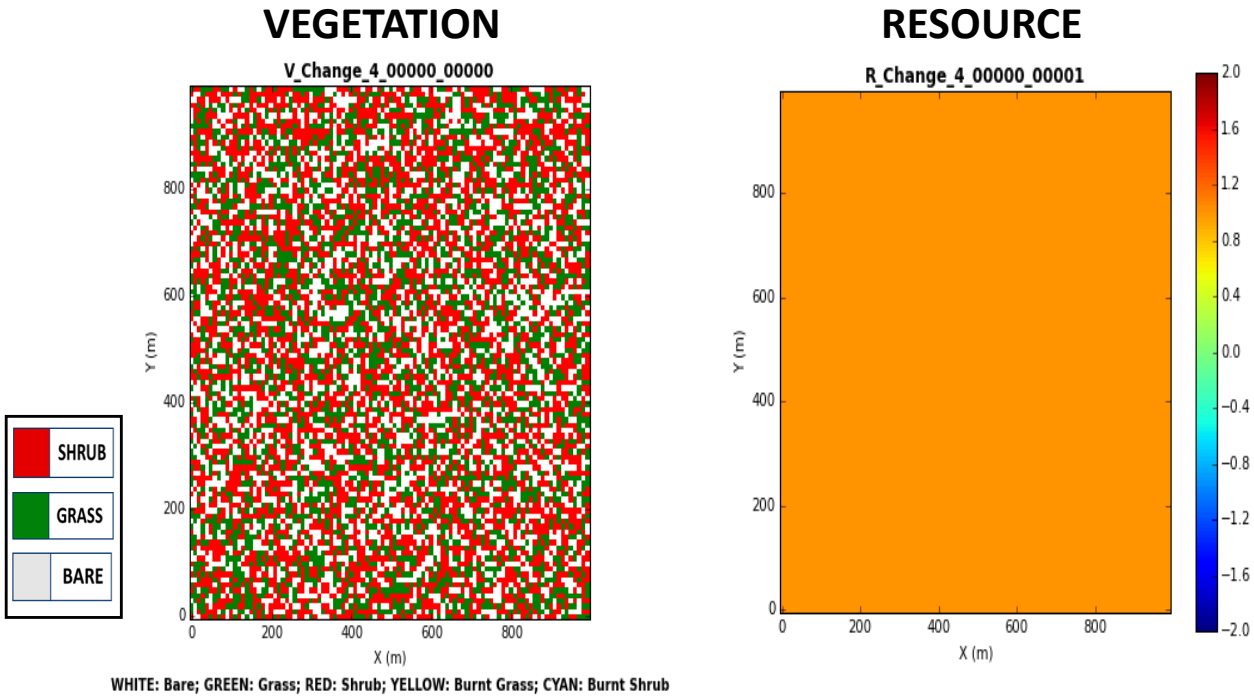


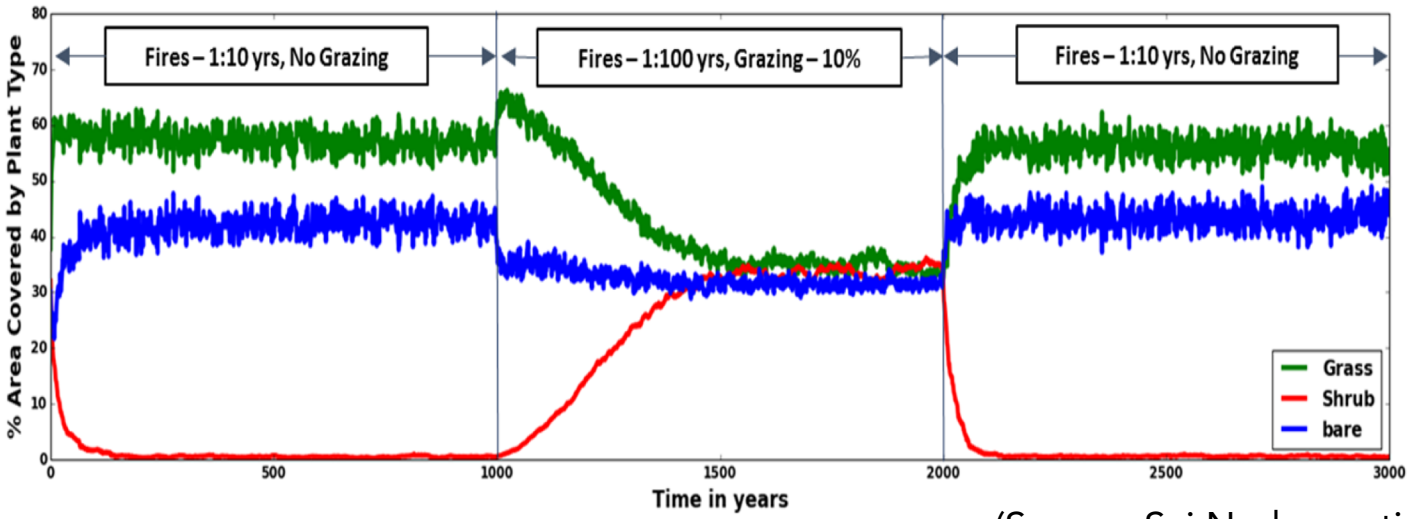
Valley widening by lateral bedrock erosion



(Source: Abby Langston, Kansas State University)

Vegetation change with climate



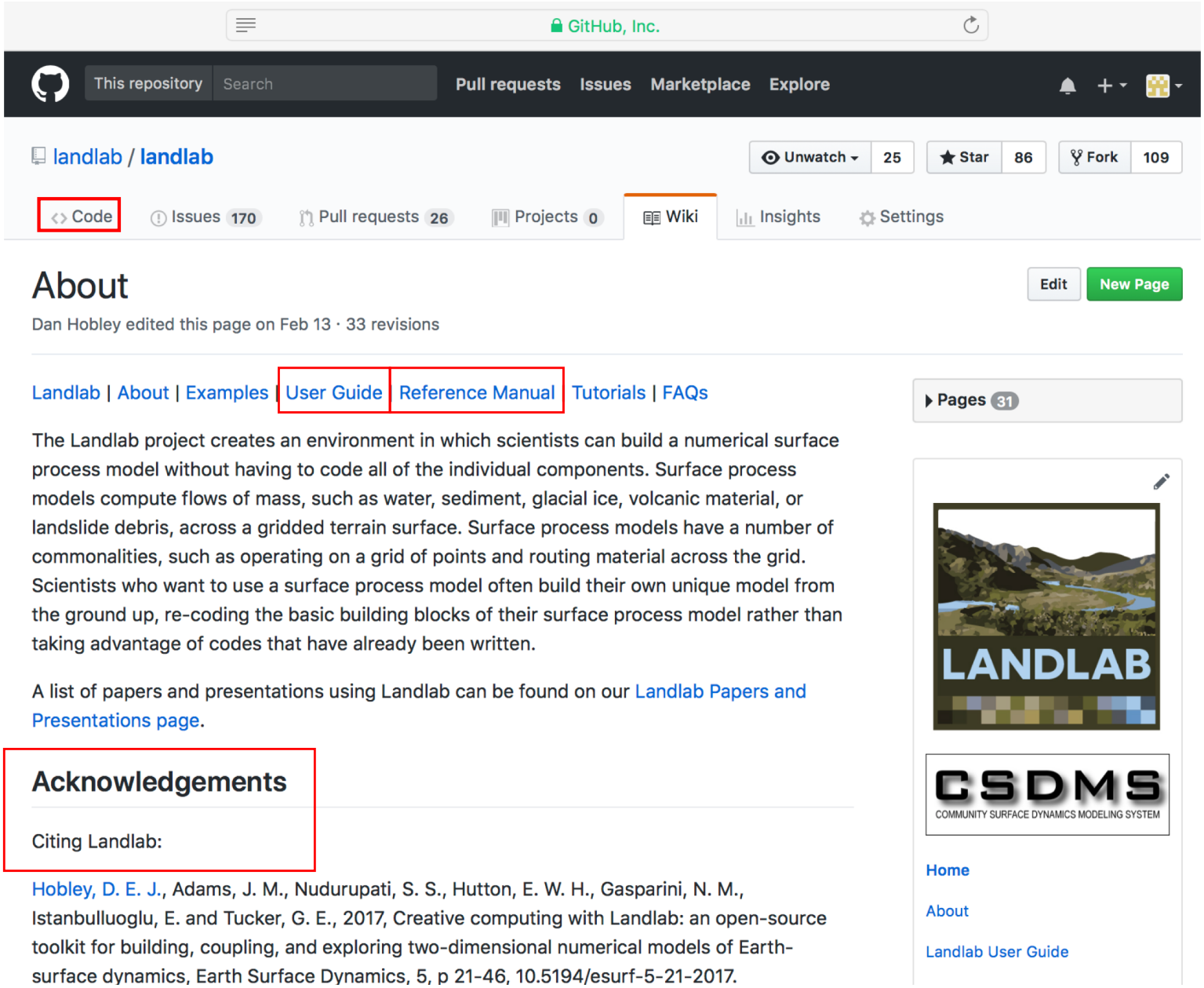


(Source: Sai Nudurupati, U. Washington)

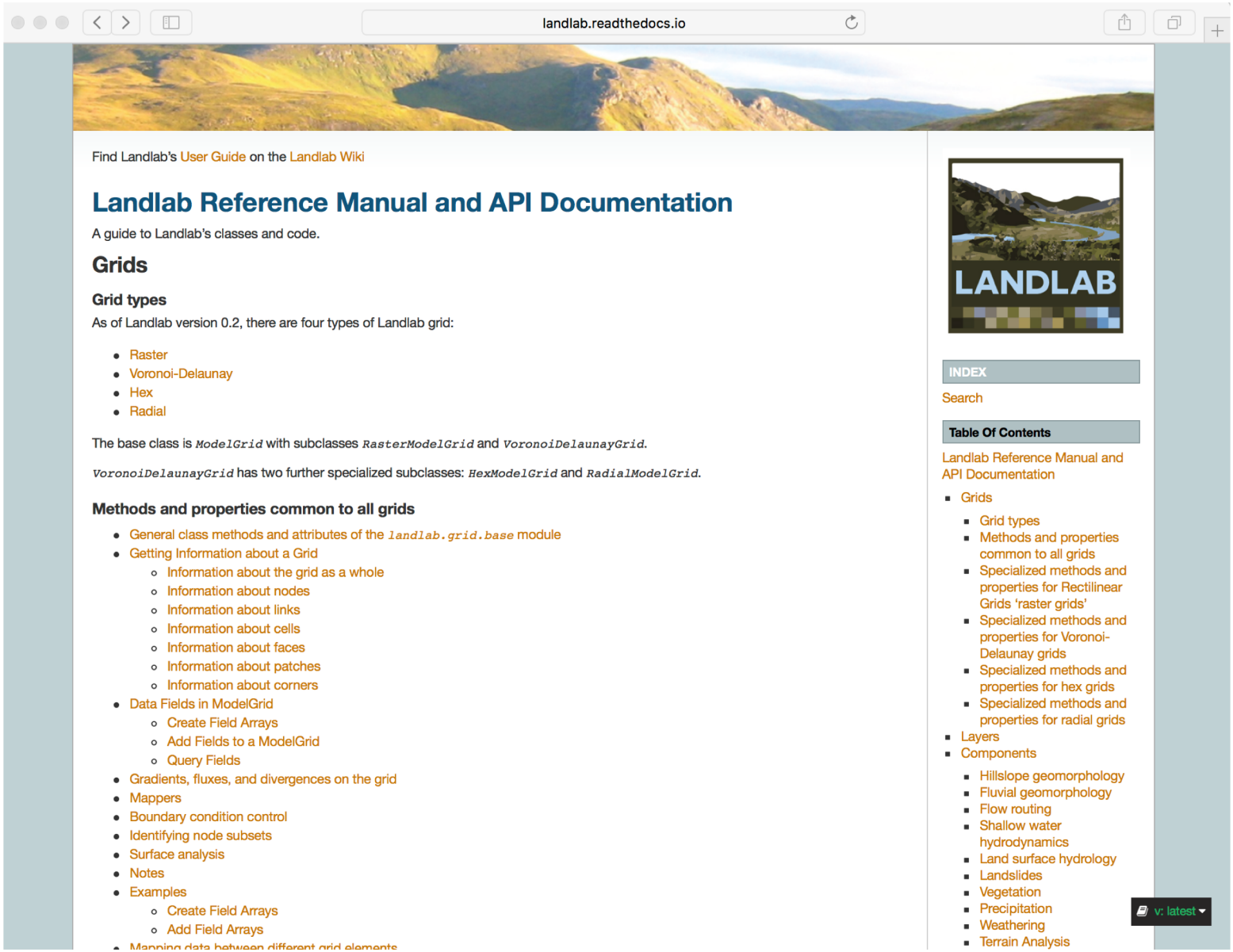
Landlab resources

landlab.github.io

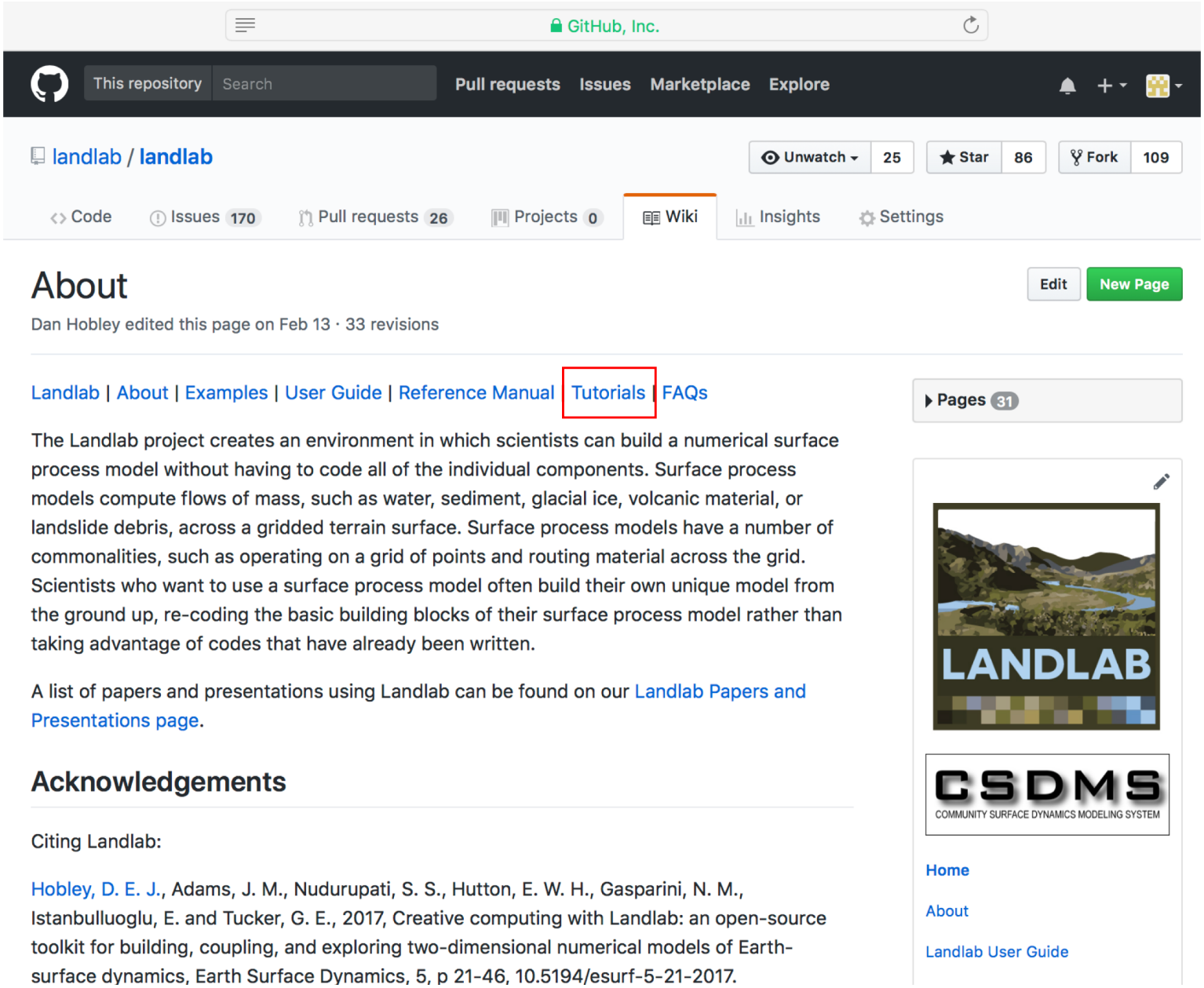
github.com/landlab/landlab/wiki



landlab.readthedocs.io



github.com/landlab/landlab/wiki

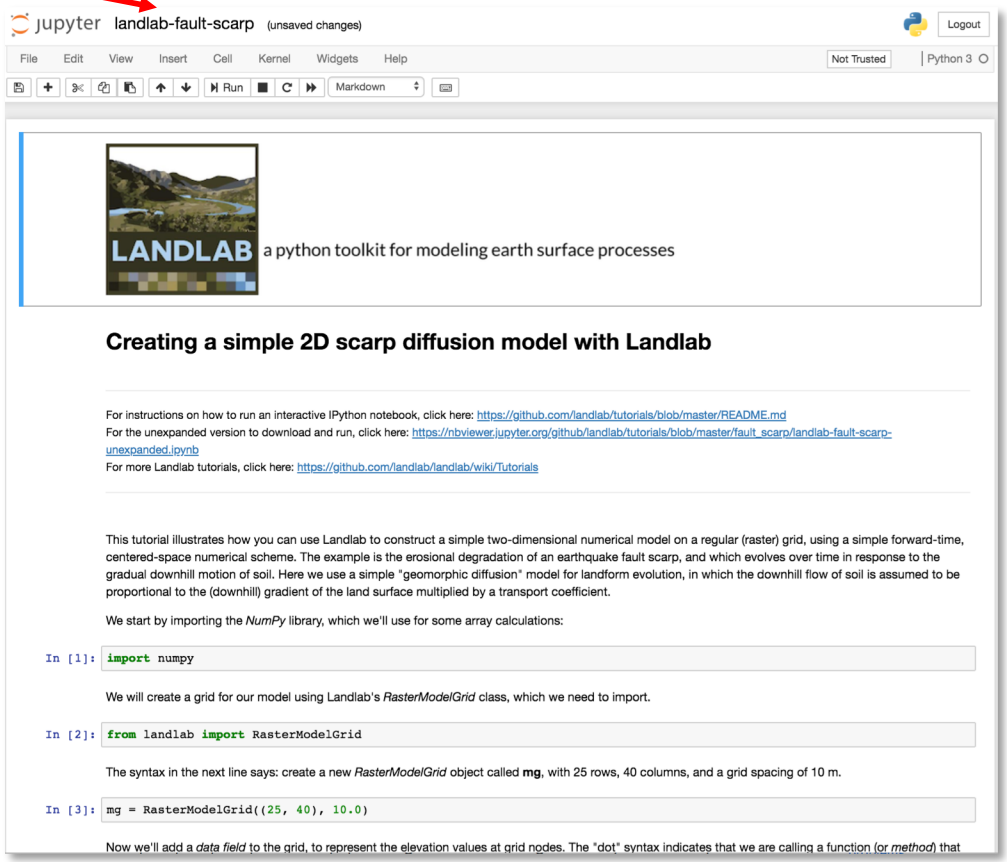


Tutorials

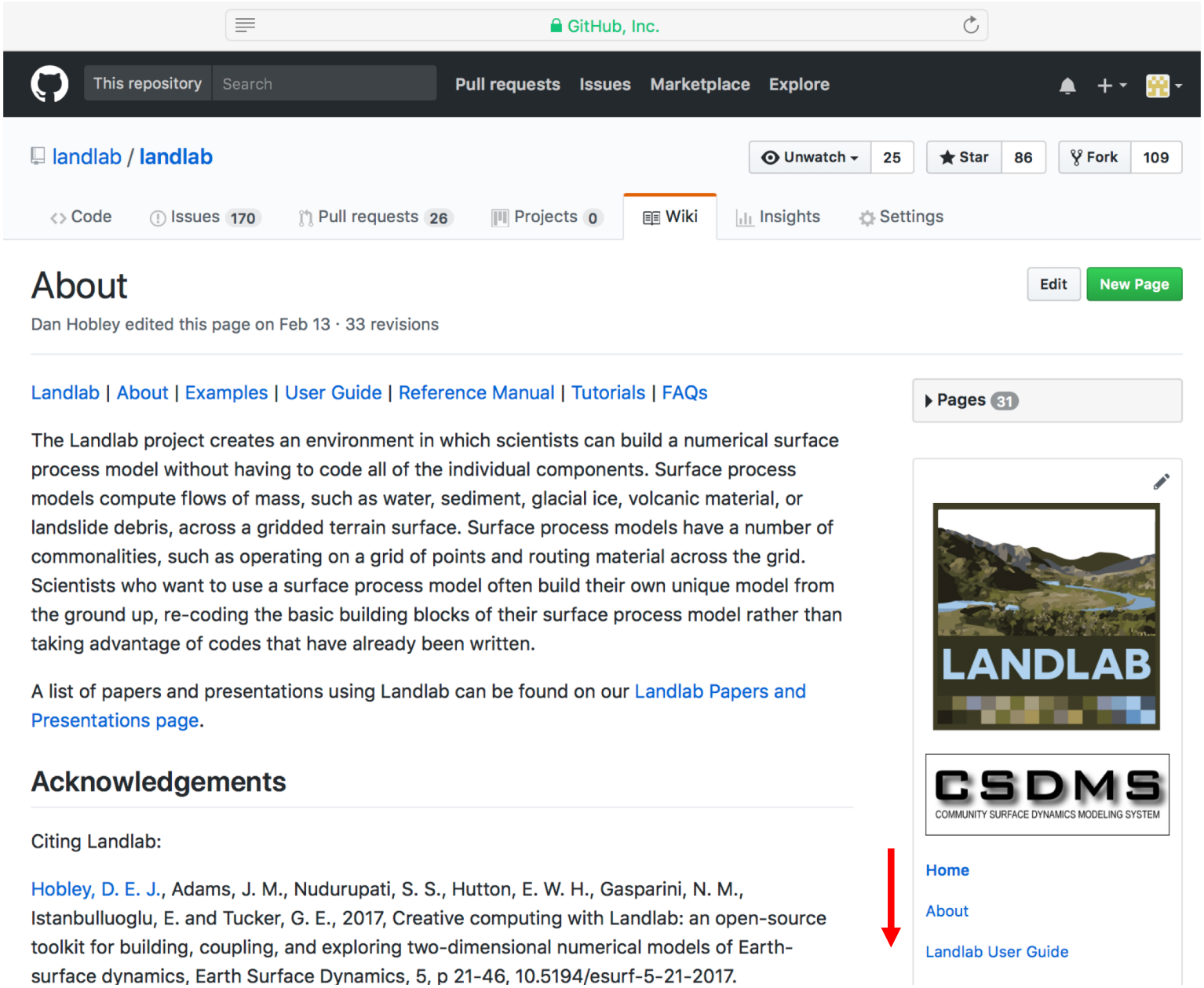
- Introduction to Python and NumPy. *Learn about:* The very basics of Python.
- Introduction to Landlab: example model of fault-scarp degradation. A short overview of some of the things Landlab can do.
- Introduction to the model grid object. Grid topology; how landlab represents data; connectivity of grid elements.
- Introduction to Landlab data fields. How Landlab stores spatial data on the grid; a little on naming conventions.
- Introduction to plotting output with Landlab. The basics of plotting with Landlab; combining matplotlib and out plots; the all-powerful `imshow_grid()` function.
- Introduction to using the Landlab component library. The basics of working with and coupling components, using *diffusion*, *stream power*, and a *storm generator* as examples.
- Using the gradient and flux-divergence functions. Landlab as solving environment for staggered grid finite difference differential approximations; functions available to help you do this.
- Mapping values from nodes to links. Options for getting data on links to nodes, nodes to links, etc.; min, max, and mean; upwinding and downwinding schemes; one-to-one, one-to-many, and many-to-one mappings.
- Setting boundary conditions on Landlab grids (several tutorials) How Landlab conceptualises boundary conditions; various ways to interact and work with them.
- Reading DEMs into Landlab Getting an ARC ESRI ASCII into Landlab; getting the boundary conditions set right.
- How to write a Landlab component What makes up a Landlab Component Standard Interface; how to make one for your process model.

Notebook tutorials on Landlab's components include:

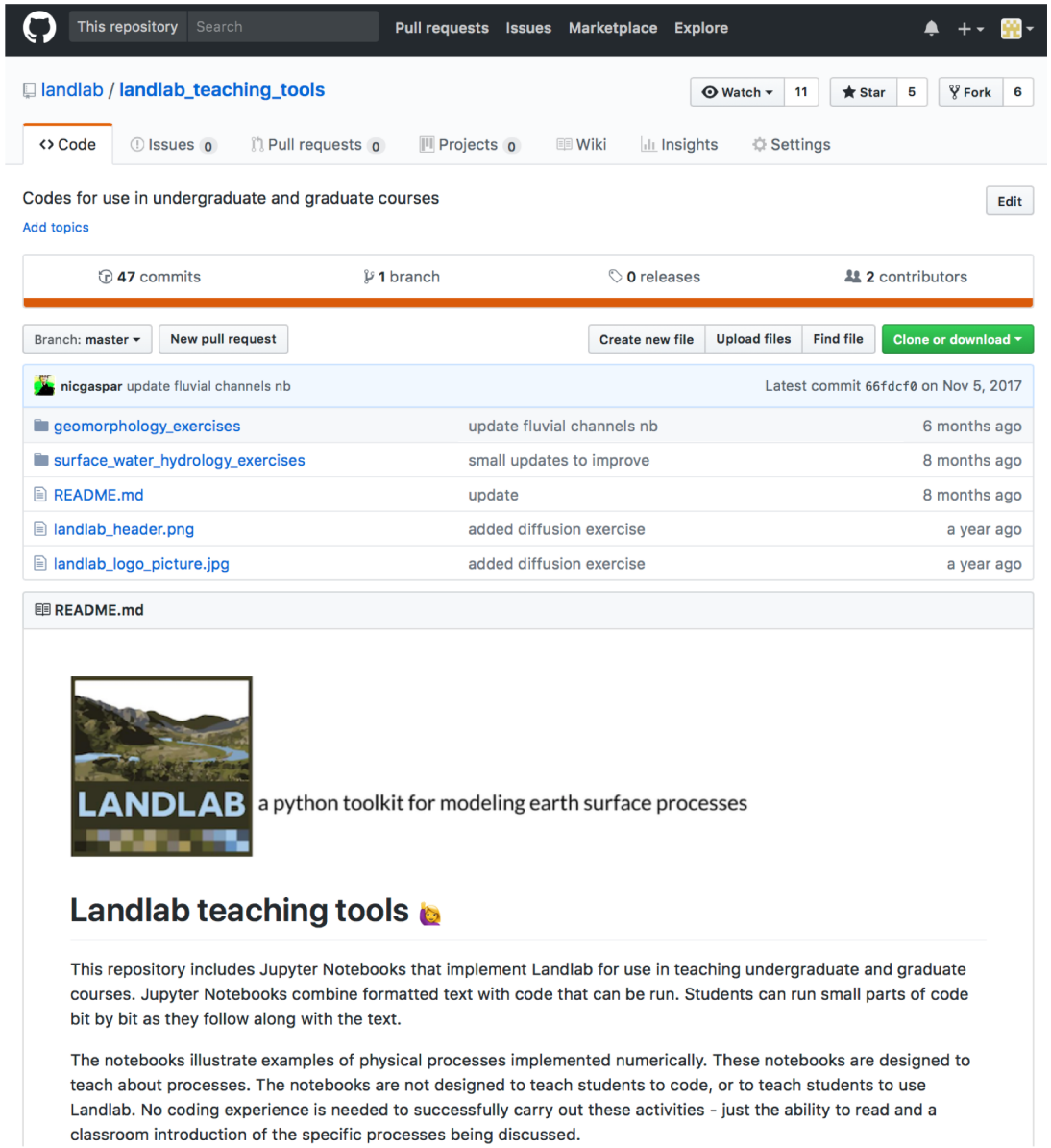
- Flow Direction and Accumulation
  - Introduction to the FlowDirector Components
  - Introduction to the FlowAccumulator Component
  - Comparison of FlowDirector Components
- Flexure
- Overland flow
- Diffusion, stream power, and the storm generator
- Ecohydrology Model on Flat Domain
- Ecohydrology Model on Actual Landscape



github.com/landlab/landlab/wiki



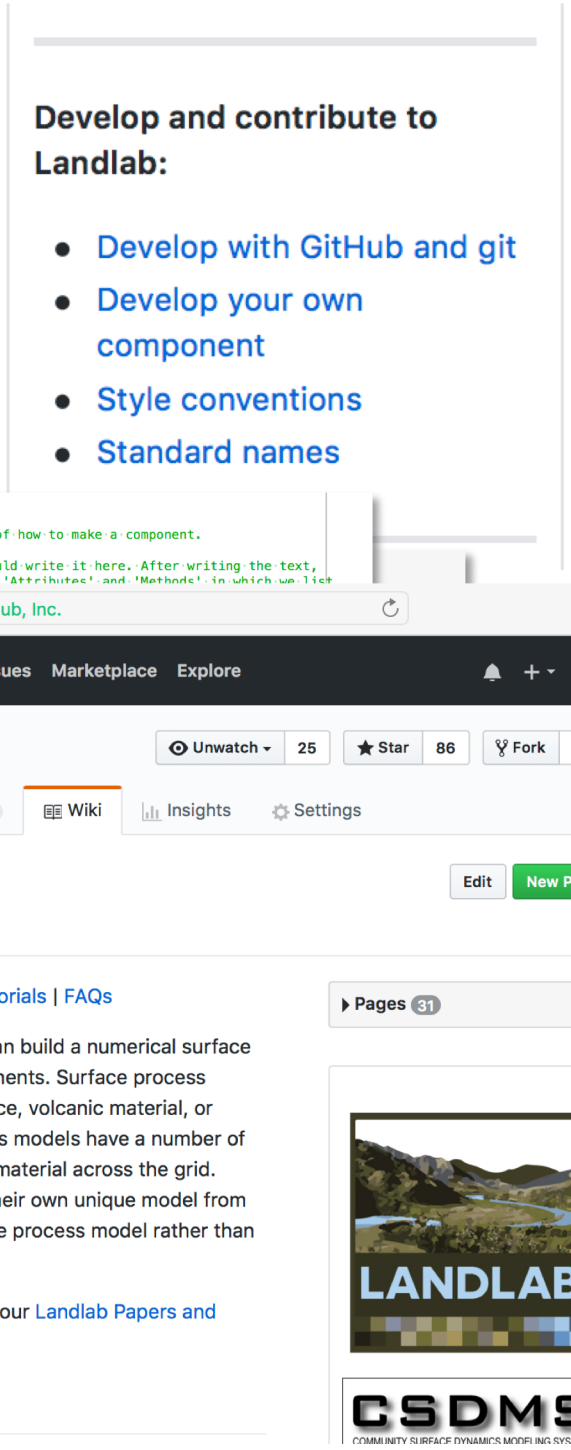
github.com/landlab/landlab_teaching_tools



Contribute to Landlab

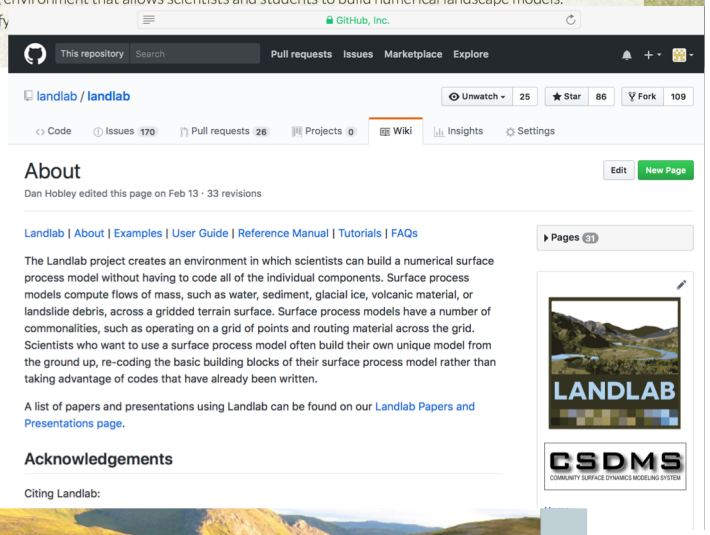
Landlab grows and improves thanks to user contributions. We encourage you to develop your own component or utility!

- wiki page + tutorial
- Template
- Contact us!

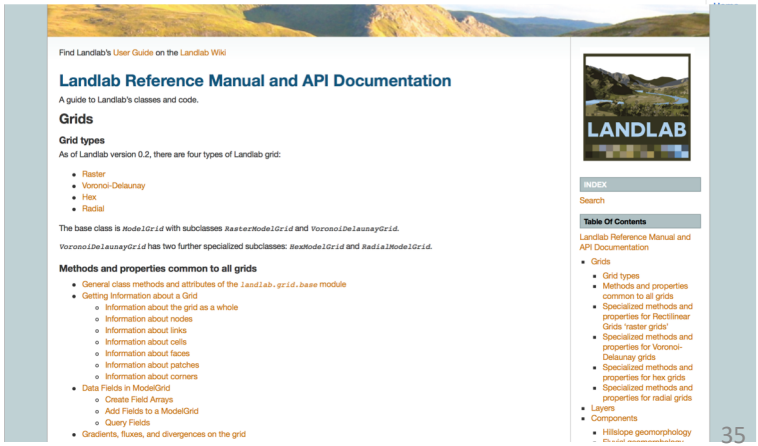


landlab.github.io

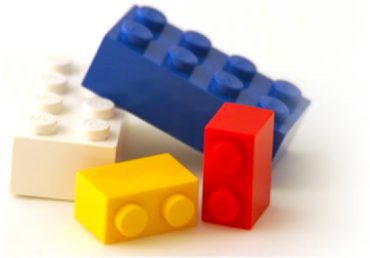
github.com/landlab/landlab/wiki



landlab.readthedocs.io



Let’s play with Landlab!



HydroShare

Go to: www.hydroshare.org