

# Modeling earth-surface dynamics with Landlab

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



Nicole Gasparini  
Nathan Lyons



Erkan Istanbuluoglu  
Sai Nudurupati  
Christina Bandaragoda  
Ronda Strauch  
Claire Beveridge



Dan Hobley



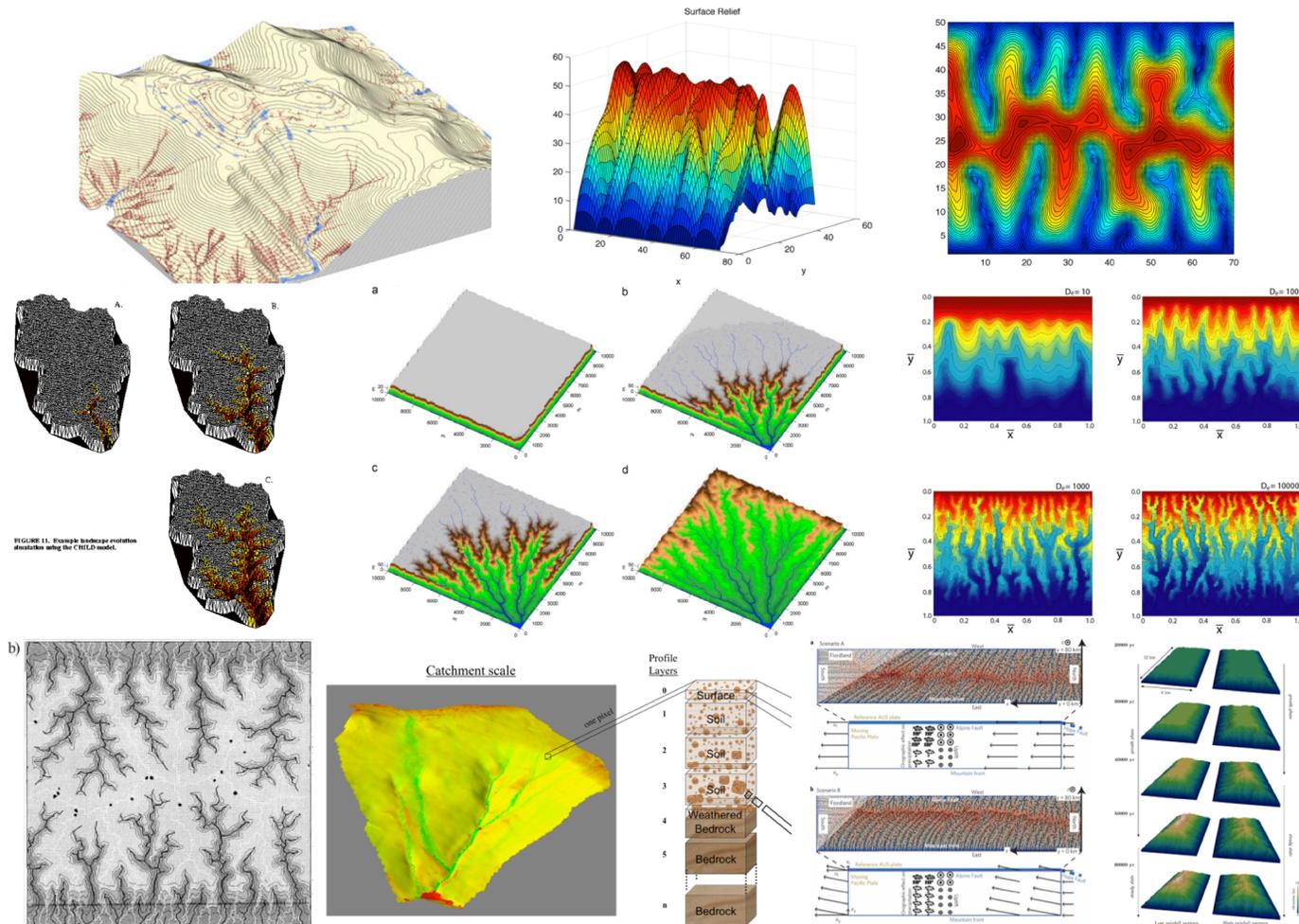
# Overview



- Introduction to Landlab
- Running Landlab on Hydroshare → [hydroshare.org](https://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:

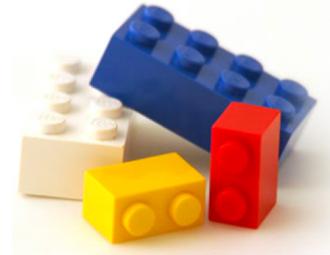
- Built for specific goals → ~~low adaptability,~~  
~~no/poor documentation and maintenance~~
- Costly (time + money) → ~~low efficiency and accessibility~~
- Closed source → ~~high cost, poor code assessment and~~  
~~reproducibility~~
- No standardization → ~~no interoperability~~
- Common elements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



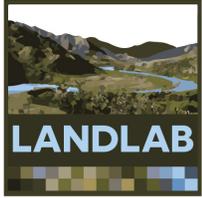
# What is Landlab?

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!

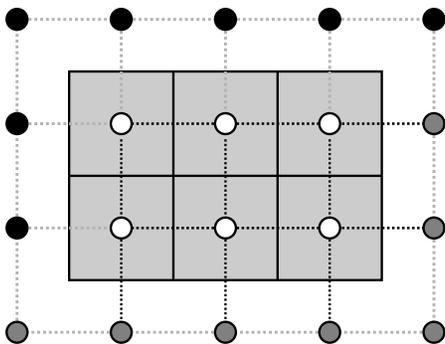
# The Landlab library



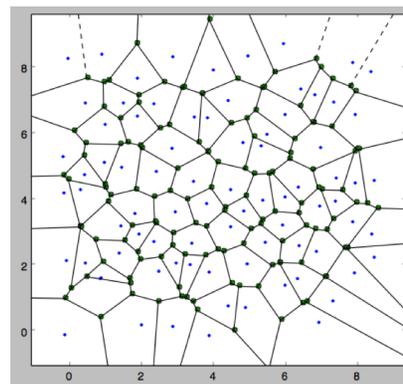
# 1. Landlab grid

- 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

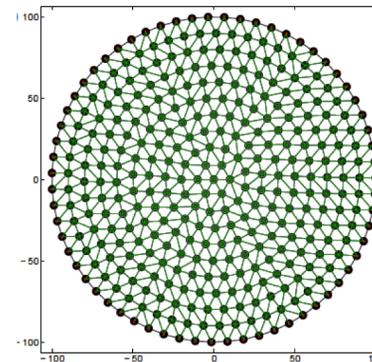
RASTER



VORONOI / DELAUNAY

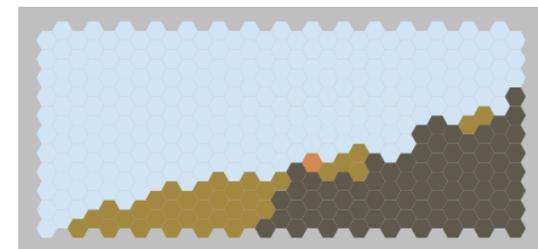


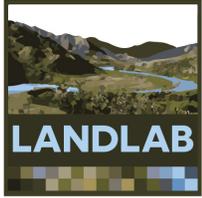
RADIAL



+ coming soon:  
Network grid

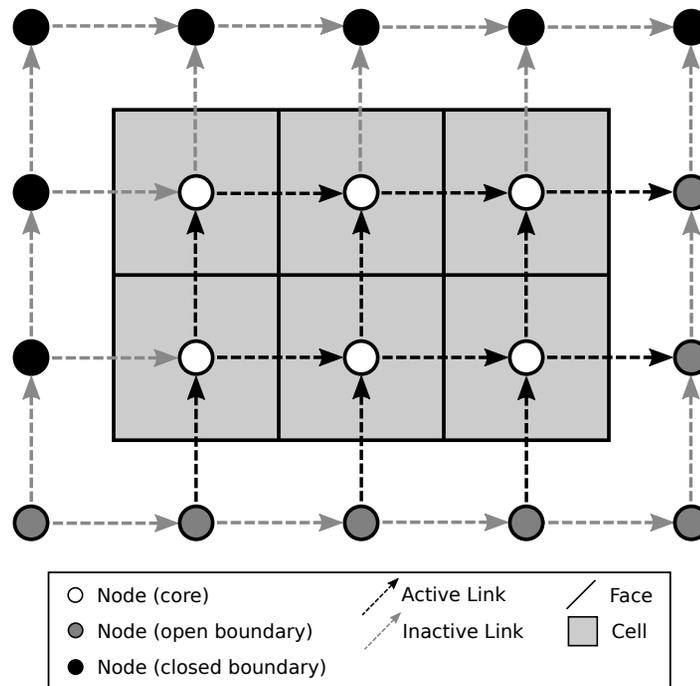
HEXAGONAL

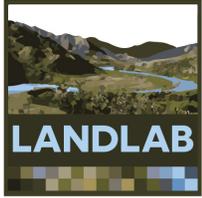




# Example: creating a grid

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

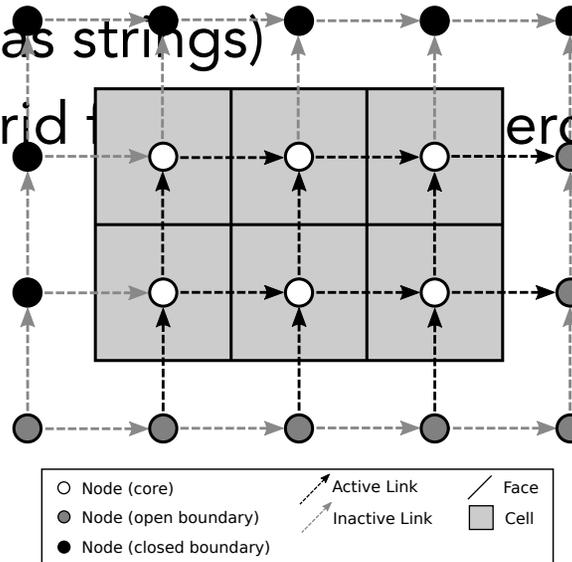


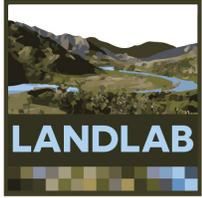


# Fields: attaching data to the grid

- 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.t`, `eros`, `add_ones`, or `add_empty`





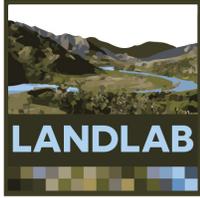
## 2. Landlab components and coupling framework for components

- 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)
```

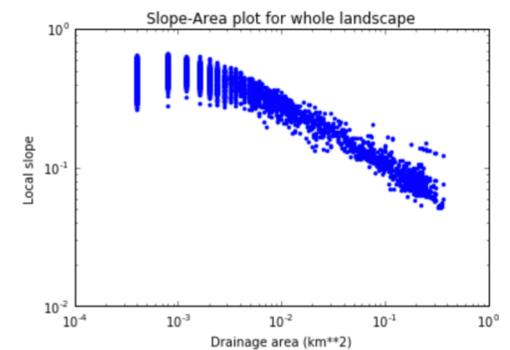
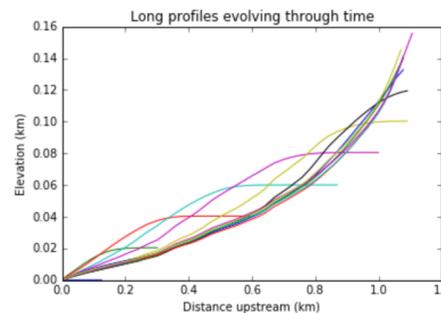
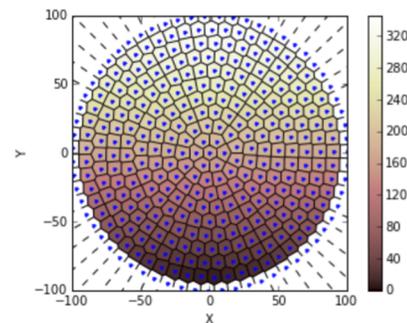
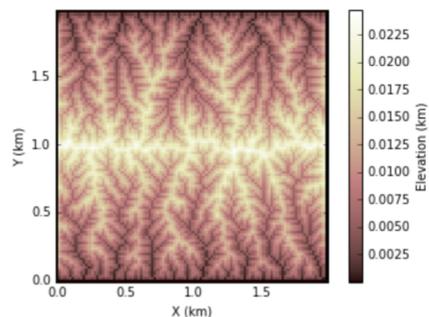
# List of components

- **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.

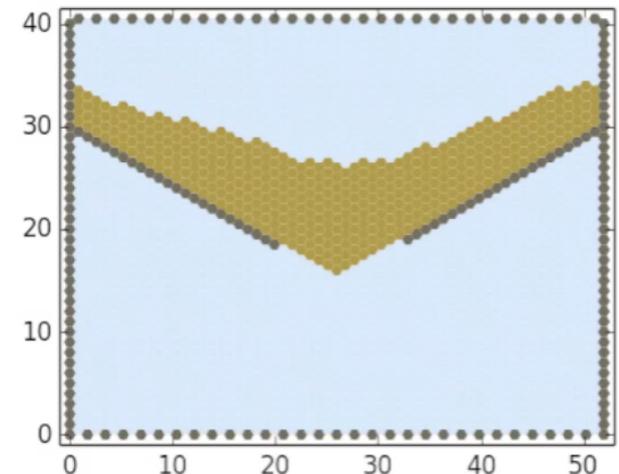
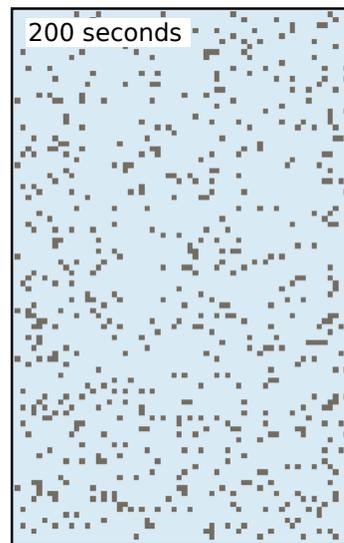
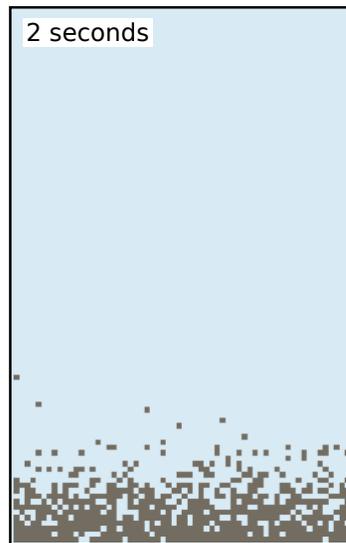
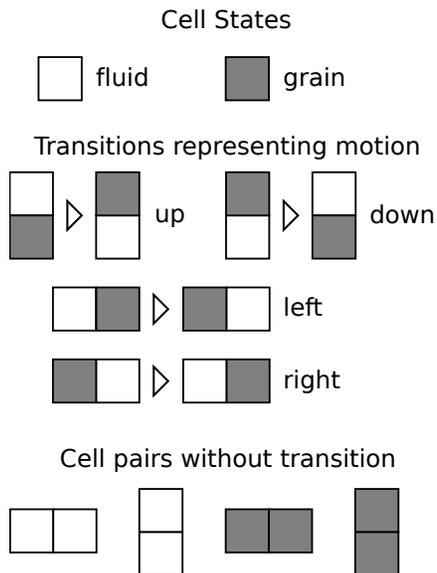




# Cellular-automaton modeling

Support for cellular-automaton modeling

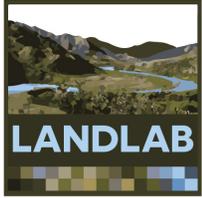
CellLab-CTS: Continuous-time stochastic CA model implementation



(Tucker et al., 2016 Geoscientific Model Development)

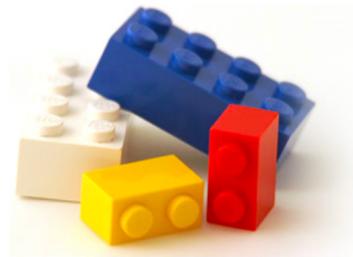
# Coming soon!

- 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

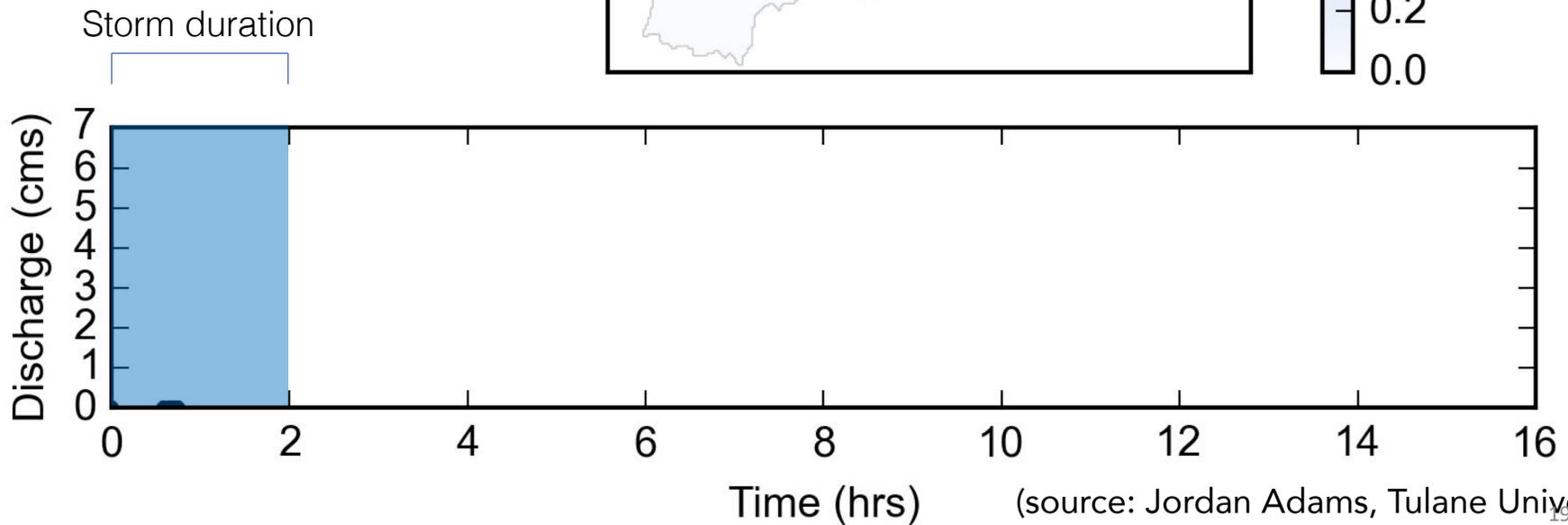
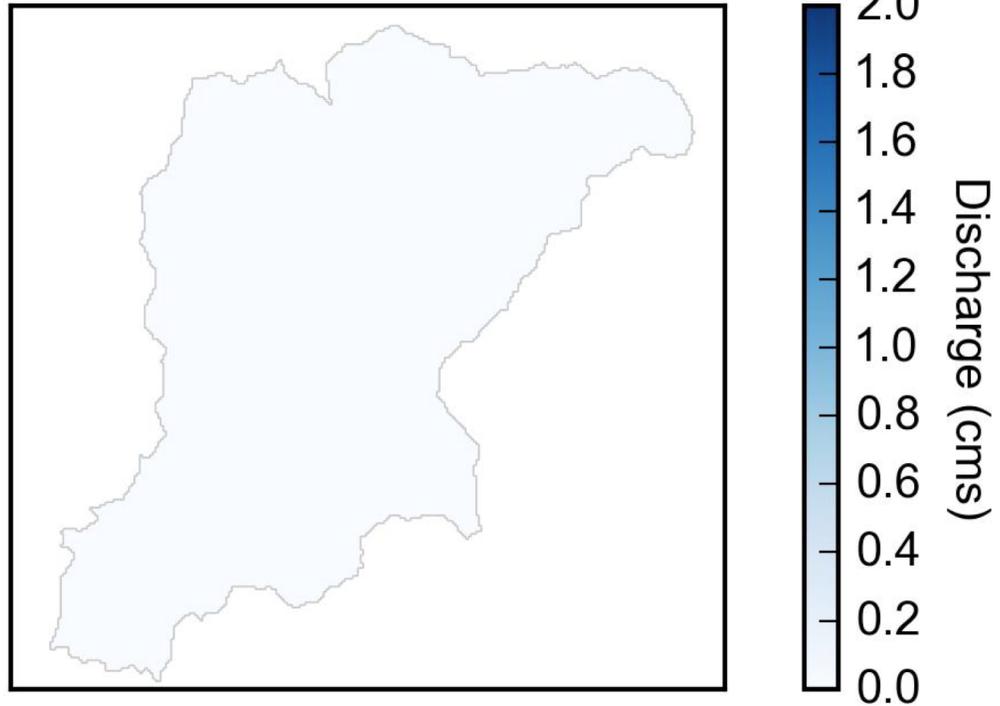
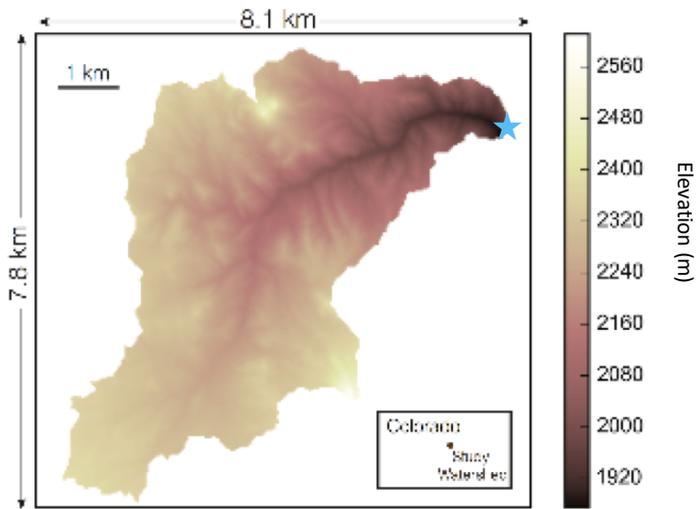
Decorators

Misc. helper functions

# Examples

# Overland Flow – hydrograph routing

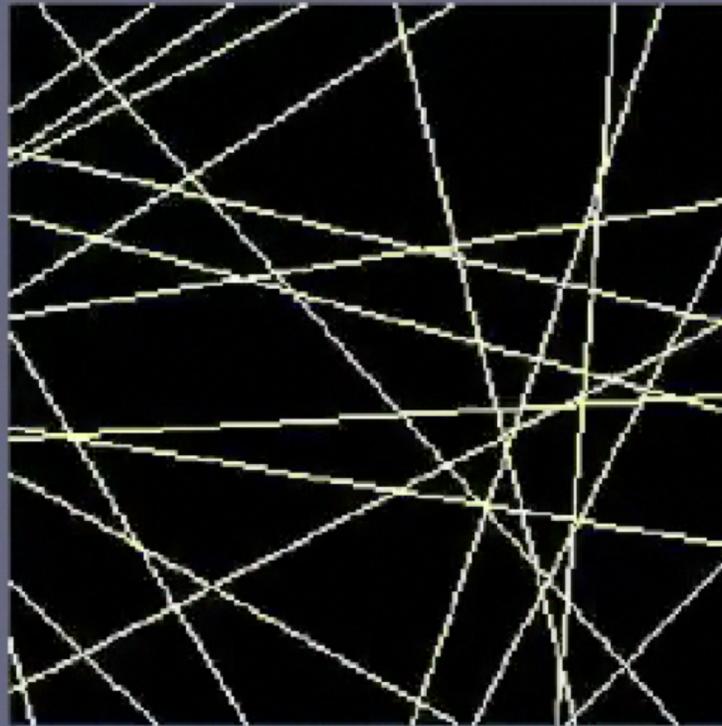
## Spring Creek, CO



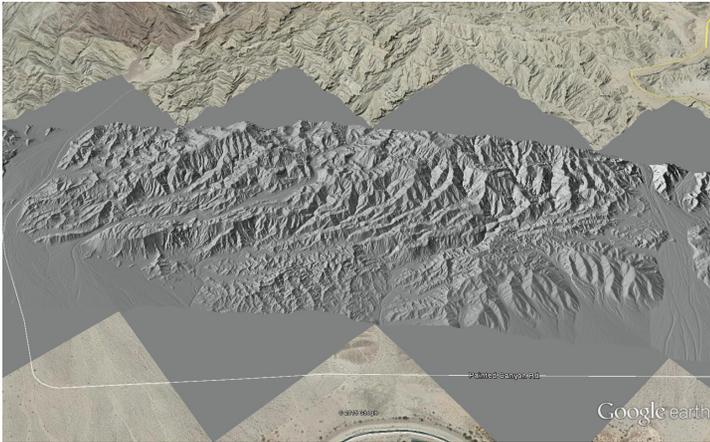
(source: Jordan Adams, Tulane University)

# Cellular automaton model of weathering along fractures

(Source: Greg Tucker, CU-Boulder)

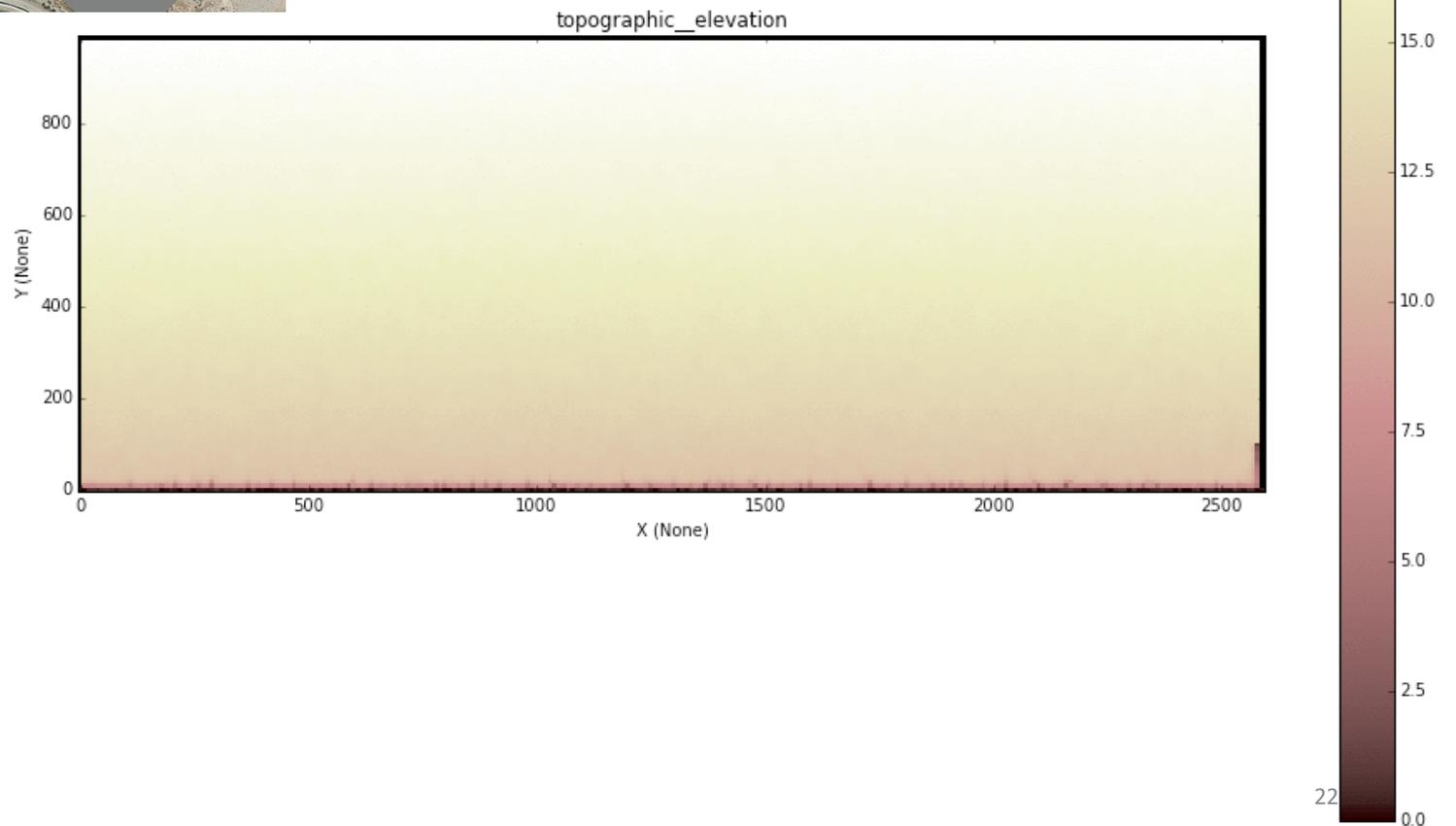


# Shear on strike-slip faults

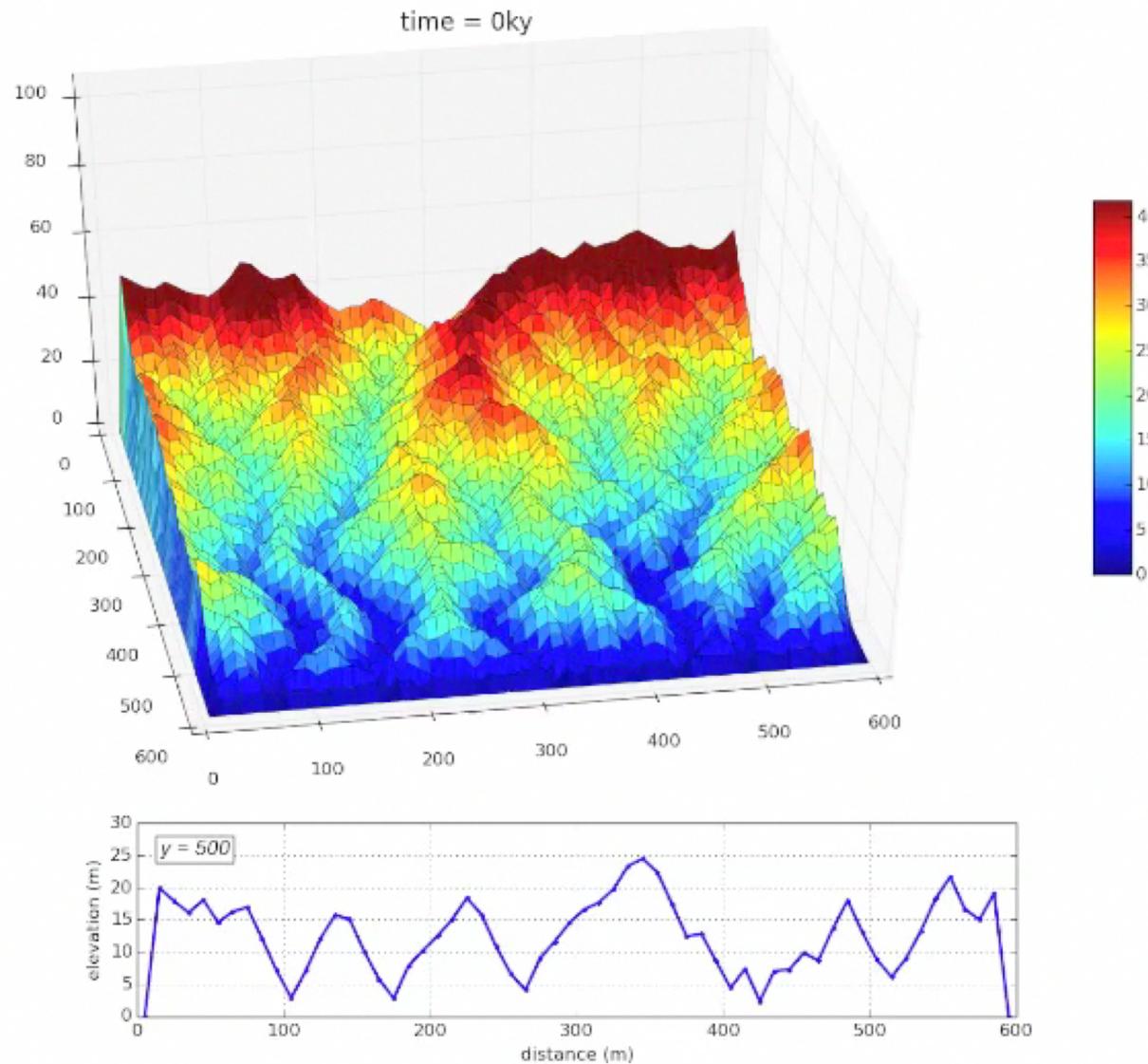


SAN ANDREAS FAULT, MECCA HILLS, CA

(Source: Harrison Gray, CU-Boulder)

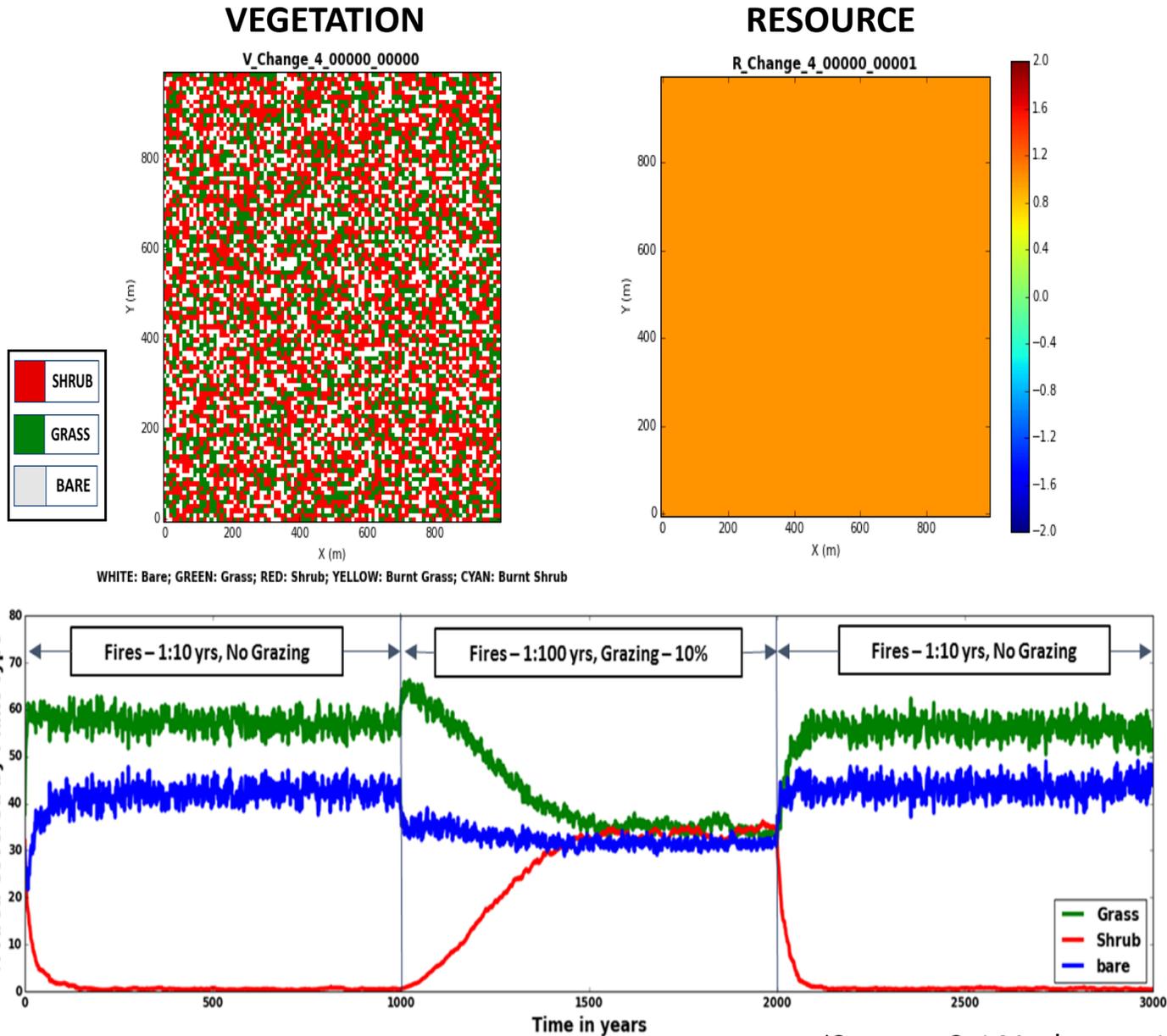


# Valley widening by lateral bedrock erosion



(Source: Abby Langston, Kansas State University)

# Vegetation change with climate



(Source: Sai Nudurupati, U. Washington)

# Landlab resources

# [landlab.github.io](https://landlab.github.io)



a python toolkit for modeling earth surface processes

[Install](#)

[User Guide](#)

[Tutorials](#)

[Reference Manual](#)

[Support](#)

[FAQs](#)

[More](#)

## What is Landlab?

Landlab is a Python-based modeling environment that allows scientists and students to build numerical landscape models. Designed for disciplines that quantify earth surface dynamics such as geomorphology, hydrology, glaciology, and stratigraphy, it can also be used in related fields.

# github.com/landlab/landlab/wiki

GitHub, Inc.

This repository Search Pull requests Issues Marketplace Explore

landlab / landlab Unwatch 25 Star 86 Fork 109

Code Issues 170 Pull requests 26 Projects 0 Wiki Insights Settings

## About

Dan Hobley edited this page on Feb 13 · 33 revisions

Landlab | About | Examples | **User Guide** | **Reference Manual** | Tutorials | FAQs

The Landlab project creates an environment in which scientists can build a numerical surface process model without having to code all of the individual components. Surface process models compute flows of mass, such as water, sediment, glacial ice, volcanic material, or landslide debris, across a gridded terrain surface. Surface process models have a number of commonalities, such as operating on a grid of points and routing material across the grid. Scientists who want to use a surface process model often build their own unique model from the ground up, re-coding the basic building blocks of their surface process model rather than taking advantage of codes that have already been written.

A list of papers and presentations using Landlab can be found on our [Landlab Papers and Presentations page](#).

### Acknowledgements

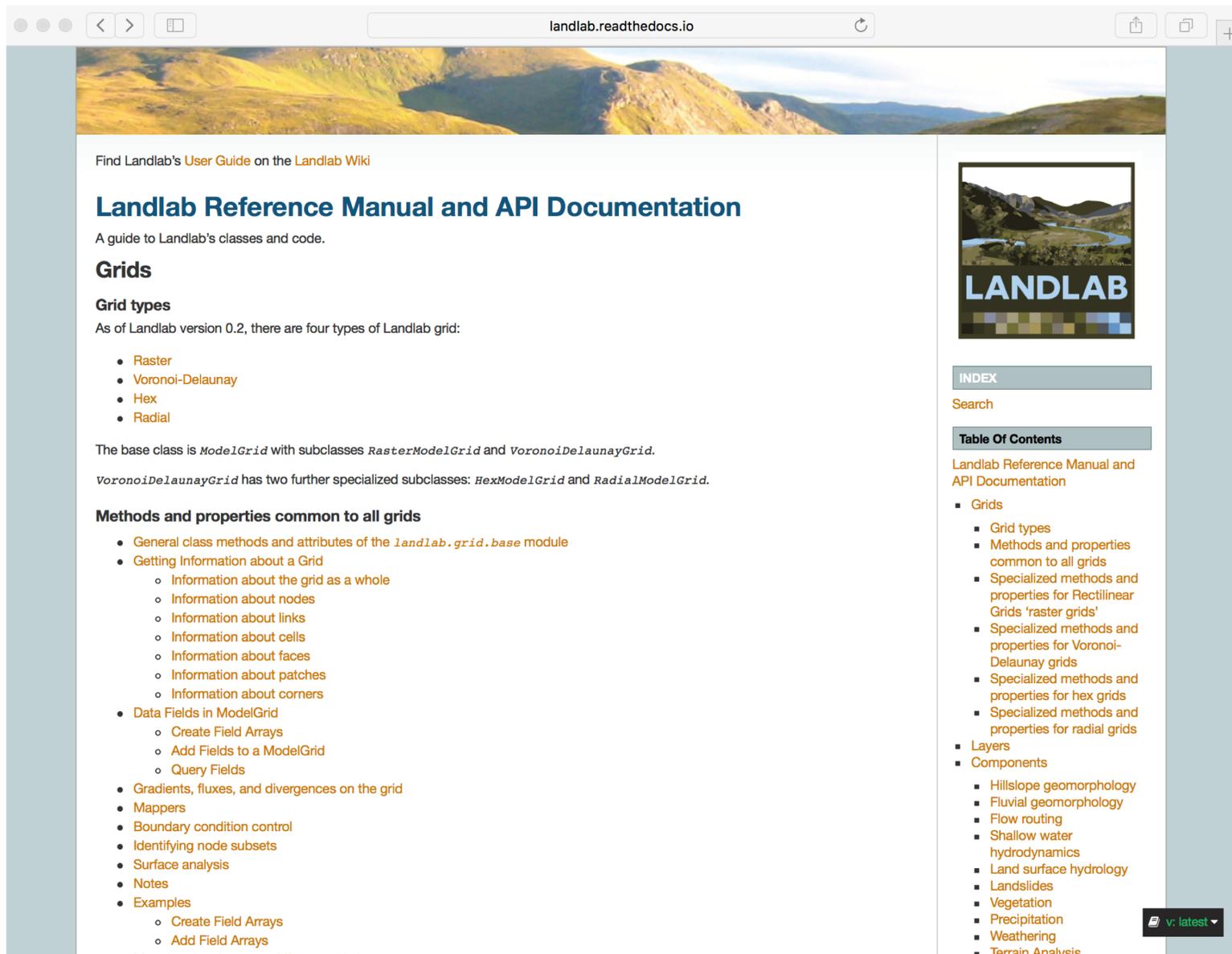
Citing Landlab:

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbuluoglu, E. and Tucker, G. E., 2017, Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics, Earth Surface Dynamics, 5, p 21-46, 10.5194/esurf-5-21-2017.

Pages 31

Home  
About  
Landlab User Guide

# [landlab.readthedocs.io](http://landlab.readthedocs.io)



Find Landlab's [User Guide](#) on the [Landlab Wiki](#)

## Landlab Reference Manual and API Documentation

A guide to Landlab's classes and code.

### Grids

#### Grid types

As of Landlab version 0.2, there are four types of Landlab grid:

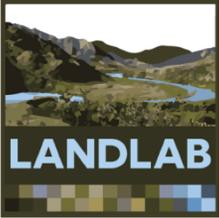
- [Raster](#)
- [Voronoi-Delaunay](#)
- [Hex](#)
- [Radial](#)

The base class is *ModelGrid* with subclasses *RasterModelGrid* and *VoronoiDelaunayGrid*.

*VoronoiDelaunayGrid* has two further specialized subclasses: *HexModelGrid* and *RadialModelGrid*.

#### Methods and properties common to all grids

- [General class methods and attributes of the `landlab.grid.base` module](#)
- [Getting Information about a Grid](#)
  - [Information about the grid as a whole](#)
  - [Information about nodes](#)
  - [Information about links](#)
  - [Information about cells](#)
  - [Information about faces](#)
  - [Information about patches](#)
  - [Information about corners](#)
- [Data Fields in ModelGrid](#)
  - [Create Field Arrays](#)
  - [Add Fields to a ModelGrid](#)
  - [Query Fields](#)
- [Gradients, fluxes, and divergences on the grid](#)
- [Mappers](#)
- [Boundary condition control](#)
- [Identifying node subsets](#)
- [Surface analysis](#)
- [Notes](#)
- [Examples](#)
  - [Create Field Arrays](#)
  - [Add Field Arrays](#)
  - [Mapping data between different grid elements](#)



INDEX

Search

#### Table Of Contents

[Landlab Reference Manual and API Documentation](#)

- [Grids](#)
  - [Grid types](#)
  - [Methods and properties common to all grids](#)
  - [Specialized methods and properties for Rectilinear Grids 'raster grids'](#)
  - [Specialized methods and properties for Voronoi-Delaunay grids](#)
  - [Specialized methods and properties for hex grids](#)
  - [Specialized methods and properties for radial grids](#)
- [Layers](#)
- [Components](#)
  - [Hillslope geomorphology](#)
  - [Fluvial geomorphology](#)
  - [Flow routing](#)
  - [Shallow water hydrodynamics](#)
  - [Land surface hydrology](#)
  - [Landslides](#)
  - [Vegetation](#)
  - [Precipitation](#)
  - [Weathering](#)
  - [Terrain Analysis](#)

v: latest

# github.com/landlab/landlab/wiki

GitHub, Inc.

This repository Search Pull requests Issues Marketplace Explore

landlab / landlab Unwatch 25 Star 86 Fork 109

Code Issues 170 Pull requests 26 Projects 0 Wiki Insights Settings

## About

Dan Hobley edited this page on Feb 13 · 33 revisions

Landlab | About | Examples | User Guide | Reference Manual | **Tutorials** | FAQs

The Landlab project creates an environment in which scientists can build a numerical surface process model without having to code all of the individual components. Surface process models compute flows of mass, such as water, sediment, glacial ice, volcanic material, or landslide debris, across a gridded terrain surface. Surface process models have a number of commonalities, such as operating on a grid of points and routing material across the grid. Scientists who want to use a surface process model often build their own unique model from the ground up, re-coding the basic building blocks of their surface process model rather than taking advantage of codes that have already been written.

A list of papers and presentations using Landlab can be found on our [Landlab Papers and Presentations page](#).

## Acknowledgements

Citing Landlab:

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbuluoglu, E. and Tucker, G. E., 2017, Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics, Earth Surface Dynamics, 5, p 21-46, 10.5194/esurf-5-21-2017.

Pages 31

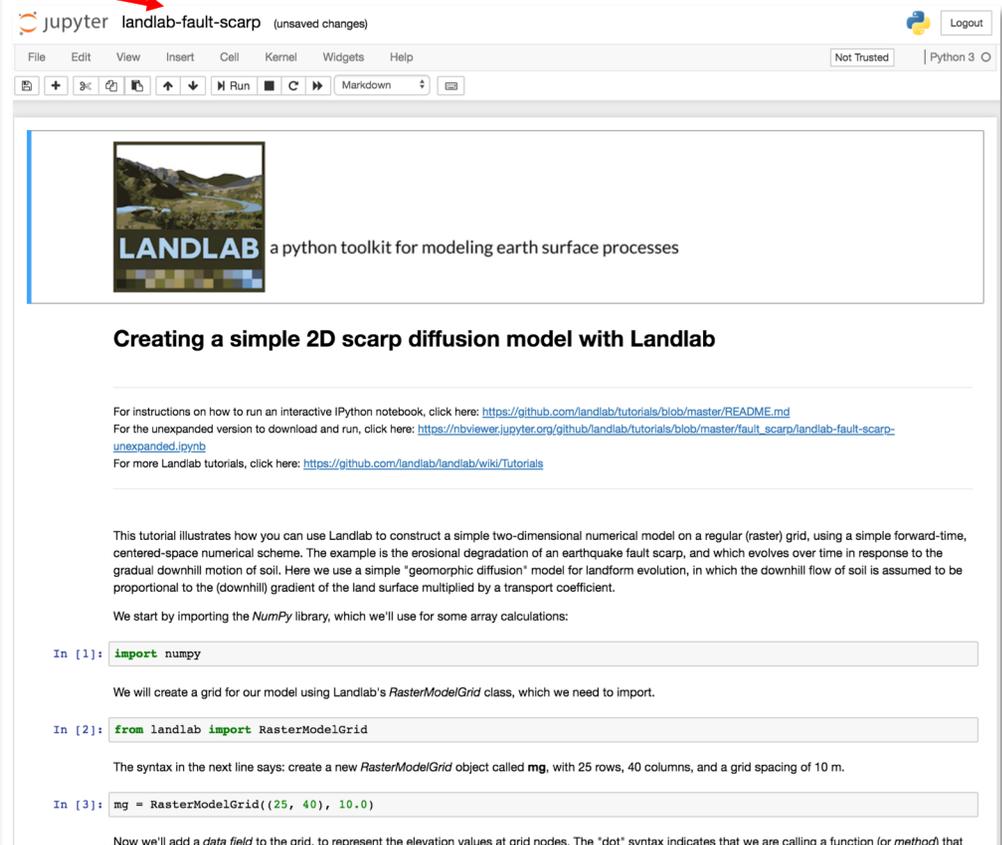
Home  
About  
Landlab User Guide

# 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



The screenshot shows a Jupyter Notebook interface. At the top, the title is "landlab-fault-scarp (unsaved changes)". Below the title bar is a menu with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The main content area displays the Landlab logo, which includes a landscape image and the text "LANDLAB a python toolkit for modeling earth surface processes". Below the logo, the notebook title "Creating a simple 2D scarp diffusion model with Landlab" is shown. The notebook content includes introductory text, links to documentation, and code cells. The first code cell contains `import numpy`. The second code cell contains `from landlab import RasterModelGrid`. The third code cell contains `mg = RasterModelGrid((25, 40), 10.0)`.

# [github.com/landlab/landlab/wiki](https://github.com/landlab/landlab/wiki)

GitHub, Inc.

This repository Search Pull requests Issues Marketplace Explore

landlab / landlab Unwatch 25 Star 86 Fork 109

Code Issues 170 Pull requests 26 Projects 0 Wiki Insights Settings

## About

Dan Hobley edited this page on Feb 13 · 33 revisions

[Landlab](#) | [About](#) | [Examples](#) | [User Guide](#) | [Reference Manual](#) | [Tutorials](#) | [FAQs](#)

The Landlab project creates an environment in which scientists can build a numerical surface process model without having to code all of the individual components. Surface process models compute flows of mass, such as water, sediment, glacial ice, volcanic material, or landslide debris, across a gridded terrain surface. Surface process models have a number of commonalities, such as operating on a grid of points and routing material across the grid. Scientists who want to use a surface process model often build their own unique model from the ground up, re-coding the basic building blocks of their surface process model rather than taking advantage of codes that have already been written.

A list of papers and presentations using Landlab can be found on our [Landlab Papers and Presentations page](#).

## Acknowledgements

Citing Landlab:

Hobley, D. E. J., Adams, J. M., Nudurupati, S. S., Hutton, E. W. H., Gasparini, N. M., Istanbuluoglu, E. and Tucker, G. E., 2017, Creative computing with Landlab: an open-source toolkit for building, coupling, and exploring two-dimensional numerical models of Earth-surface dynamics, *Earth Surface Dynamics*, 5, p 21-46, 10.5194/esurf-5-21-2017.

Pages 31

Home  
About  
Landlab User Guide

# github.com/landlab/landlab\_teaching\_tools

This repository

Pull requests Issues Marketplace Explore

landlab / landlab\_teaching\_tools

Watch 11 Star 5 Fork 6

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Codes for use in undergraduate and graduate courses [Edit](#)

[Add topics](#)

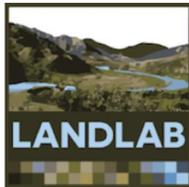
47 commits 1 branch 0 releases 2 contributors

Branch: master New pull request

Create new file Upload files Find file Clone or download

Commit	Message	Time
nicgaspar update fluvial channels nb	update fluvial channels nb	6 months ago
update fluvial channels nb	small updates to improve	8 months ago
update	update	8 months ago
added diffusion exercise	added diffusion exercise	a year ago
added diffusion exercise	added diffusion exercise	a year ago

README.md



LANDLAB a python toolkit for modeling earth surface processes

## Landlab teaching tools 🧑🎓

This repository includes Jupyter Notebooks that implement Landlab for use in teaching undergraduate and graduate courses. Jupyter Notebooks combine formatted text with code that can be run. Students can run small parts of code bit by bit as they follow along with the text.

The notebooks illustrate examples of physical processes implemented numerically. These notebooks are designed to teach about processes. The notebooks are not designed to teach students to code, or to teach students to use Landlab. No coding experience is needed to successfully carry out these activities - just the ability to read and a classroom introduction of the specific processes being discussed.

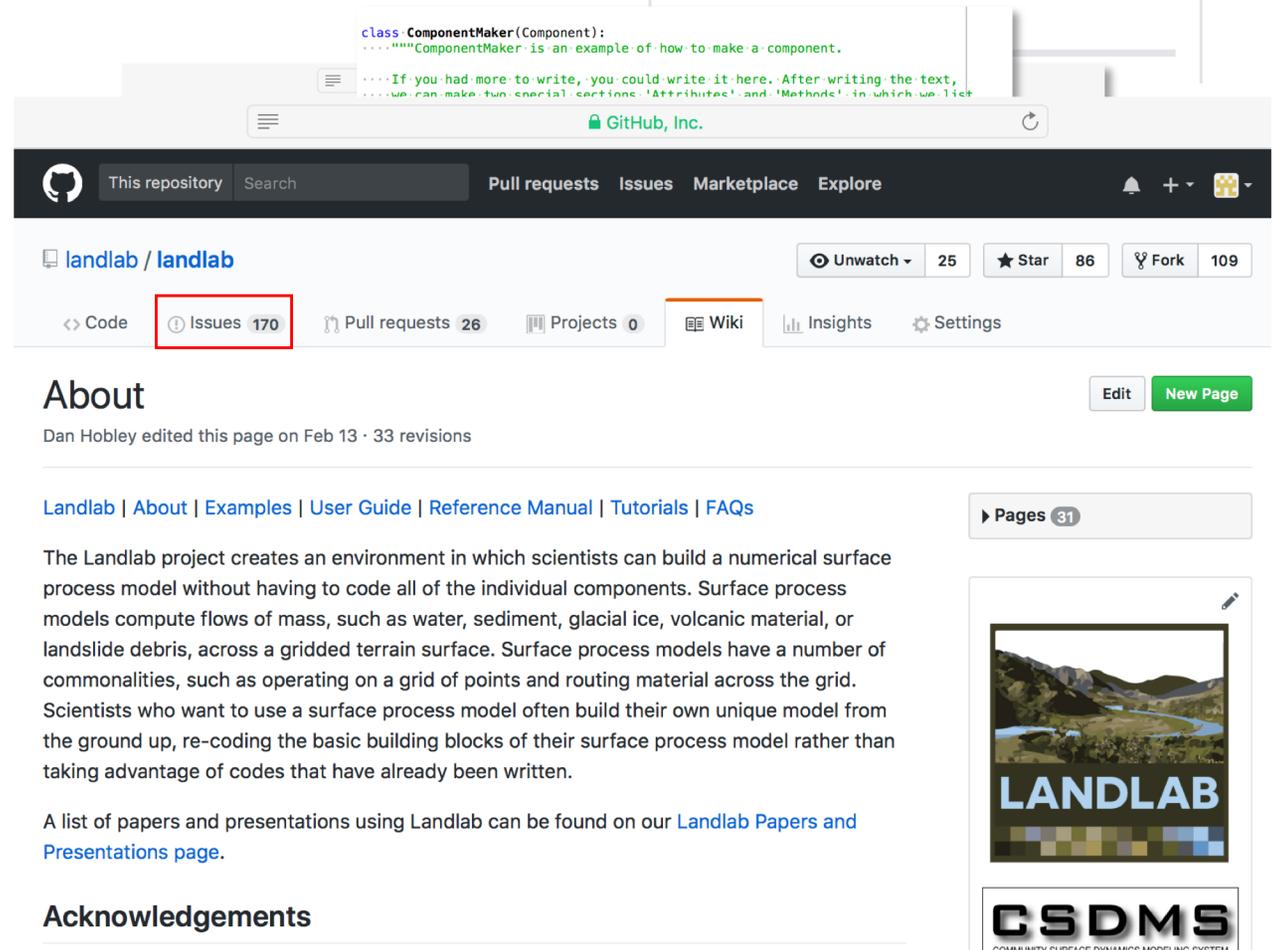
# Contribute to Landlab

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

Develop and contribute to Landlab:

- [Develop with GitHub and git](#)
- [Develop your own component](#)
- [Style conventions](#)
- [Standard names](#)

- [wiki page + tutorial](#)
- [Template](#)
- [Contact us!](#)



The screenshot shows the GitHub repository page for Landlab. The 'Issues' tab is highlighted with a red box, indicating 170 issues. The page includes navigation links for Code, Pull requests (26), Projects (0), Wiki, Insights, and Settings. The 'About' section is visible, showing the repository name 'landlab / landlab' and the number of stars (86) and forks (109). The 'About' section includes a list of links: Landlab | About | Examples | User Guide | Reference Manual | Tutorials | FAQs. The main content area contains a paragraph describing the Landlab project and a link to a list of papers and presentations. The 'Acknowledgements' section is also visible. On the right side, there is a 'Pages' section with 31 pages and a 'New Page' button. At the bottom right, there is a logo for LANDLAB and CSDMS (COMMUNITY SURFACE DYNAMICS MODELING SYSTEM).

[landlab.github.io](https://landlab.github.io)

LANDLAB a python toolkit for modeling earth surface processes

Install User Guide Tutorials Reference Manual Support FAQs More

### What is Landlab?

Landlab is a Python-based modeling environment that allows scientists and students to build numerical landscape models. Designed for disciplines that quantify can also be used in related fields.

landlab / landlab 25 86 109

About

Landlab | About | Examples | User Guide | Reference Manual | Tutorials | FAQs

The Landlab project creates an environment in which scientists can build a numerical surface process model without having to code all of the individual components. Surface process models compute flows of mass, such as water, sediment, glacial ice, volcanic material, or landslide debris, across a gridded terrain surface. Surface process models have a number of commonalities, such as operating on a grid of points and routing material across the grid. Scientists who want to use a surface process model often build their own unique model from the ground up, re-coding the basic building blocks of their surface process model rather than taking advantage of codes that have already been written.

A list of papers and presentations using Landlab can be found on our [Landlab Papers and Presentations page](#).

### Acknowledgements

Citing Landlab:

[github.com/landlab/landlab/wiki](https://github.com/landlab/landlab/wiki)

[landlab.readthedocs.io](https://landlab.readthedocs.io)

## Landlab Reference Manual and API Documentation

A guide to Landlab's classes and code.

### Grids

As of Landlab version 0.2, there are four types of Landlab grid:

- Raster
- Voronoi-Delaunay
- Hex
- Radial

The base class is `ModelGrid` with subclasses `RasterModelGrid` and `VoronoiDelaunayGrid`. `VoronoiDelaunayGrid` has two further specialized subclasses: `HexModelGrid` and `RadialModelGrid`.

### Methods and properties common to all grids

- General class methods and attributes of the `landlab.grid.base` module
- Getting information about a Grid
  - Information about the grid as a whole
  - Information about nodes
  - Information about links
  - Information about cells
  - Information about faces
  - Information about patches
  - Information about corners
- Data Fields in `ModelGrid`
  - Create Field Arrays
  - Add Fields to a `ModelGrid`
  - Query Fields
- Gradients, fluxes, and divergences on the grid

INDEX

Search

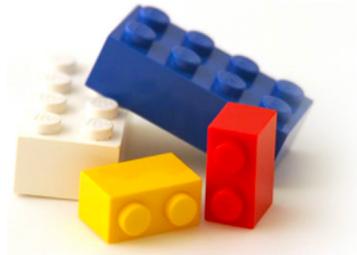
### Table Of Contents

Landlab Reference Manual and API Documentation

- Grids
  - Grid types
  - Methods and properties common to all grids
  - Specialized methods and properties for Rectilinear Grids (raster grid)
  - Specialized methods and properties for Voronoi-Delaunay grids
  - Specialized methods and properties for hex grids
  - Specialized methods and properties for radial grids
- Layers
- Components
  - Hillslope geomorphology

35

Let's play with Landlab!



# HydroShare

Go to: [www.hydroshare.org](http://www.hydroshare.org)



MY RESOURCES

DISCOVER

COLLABORATE

APPS

HELP

ABOUT

→ SIGN IN

## Join the community to start sharing

HydroShare is an online collaboration environment for sharing data, models, and code.

Sign up now

## Share your data and models with colleagues

← Upload, share, and access a broad set of hydrologic data types and models. Manage who has access to the content that you share. →





