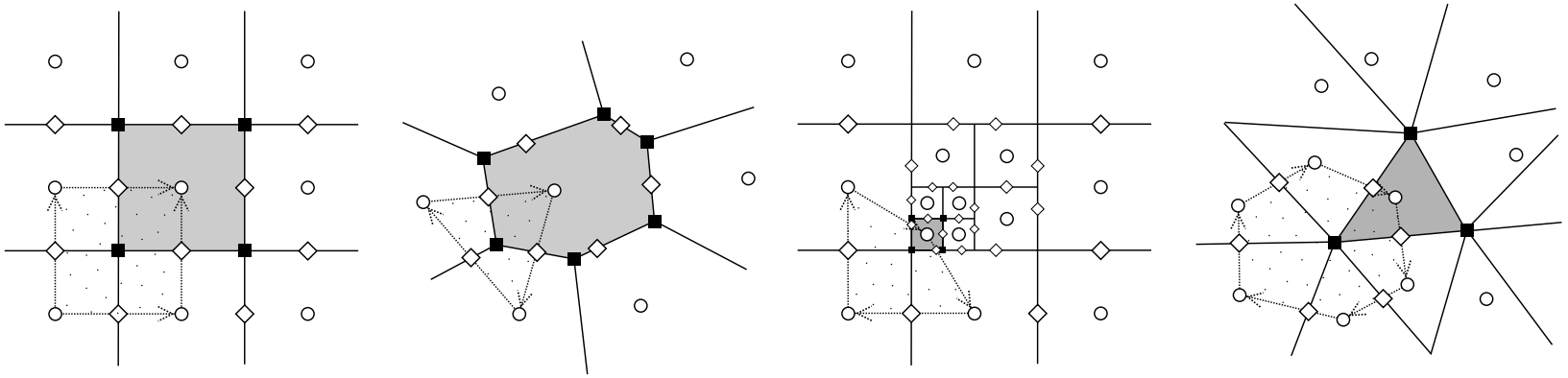
Landlab:
A new, open-source, modular, Python-based tool for modeling surface dynamics



(a) Raster grid

(b) Voronoi cells with Delaunay triangulated nodes

(c) Quadtree grid

(d) Delaunay triangle cells with Voronoi polygon patches

Daniel E.J. Hobley, Gregory E. Tucker, Jordan M. Adams, Nicole M. Gasparini, Eric Hutton, Erkan Istanbulluoglu, Sai Siddhartha

NSF

CIRES

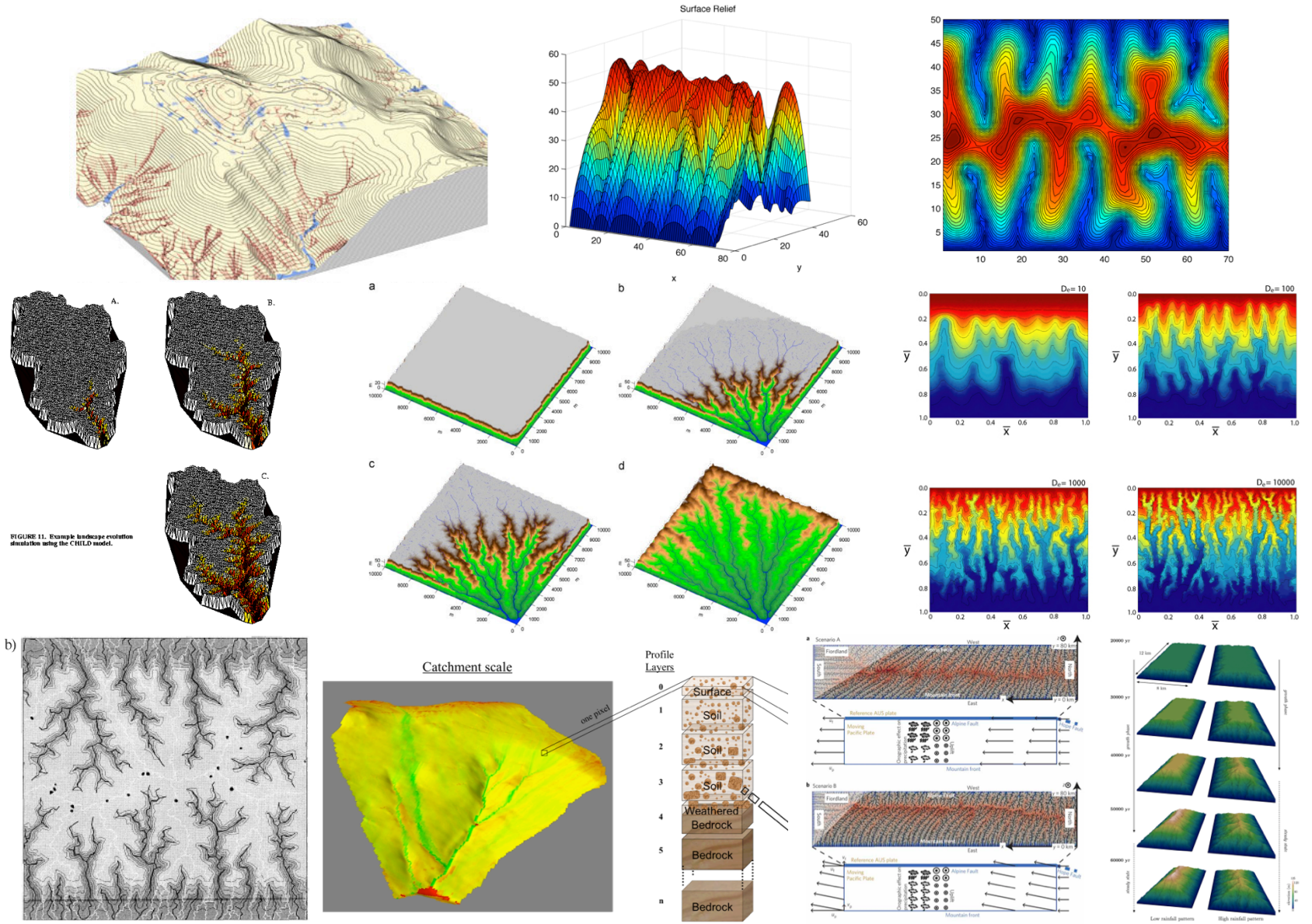
CSDMS
COMMUNITY SURFACE DYNAMICS MODELING SYSTEM

University of Colorado Boulder

Tulane University

UNIVERSITY of WASHINGTON

Motivation



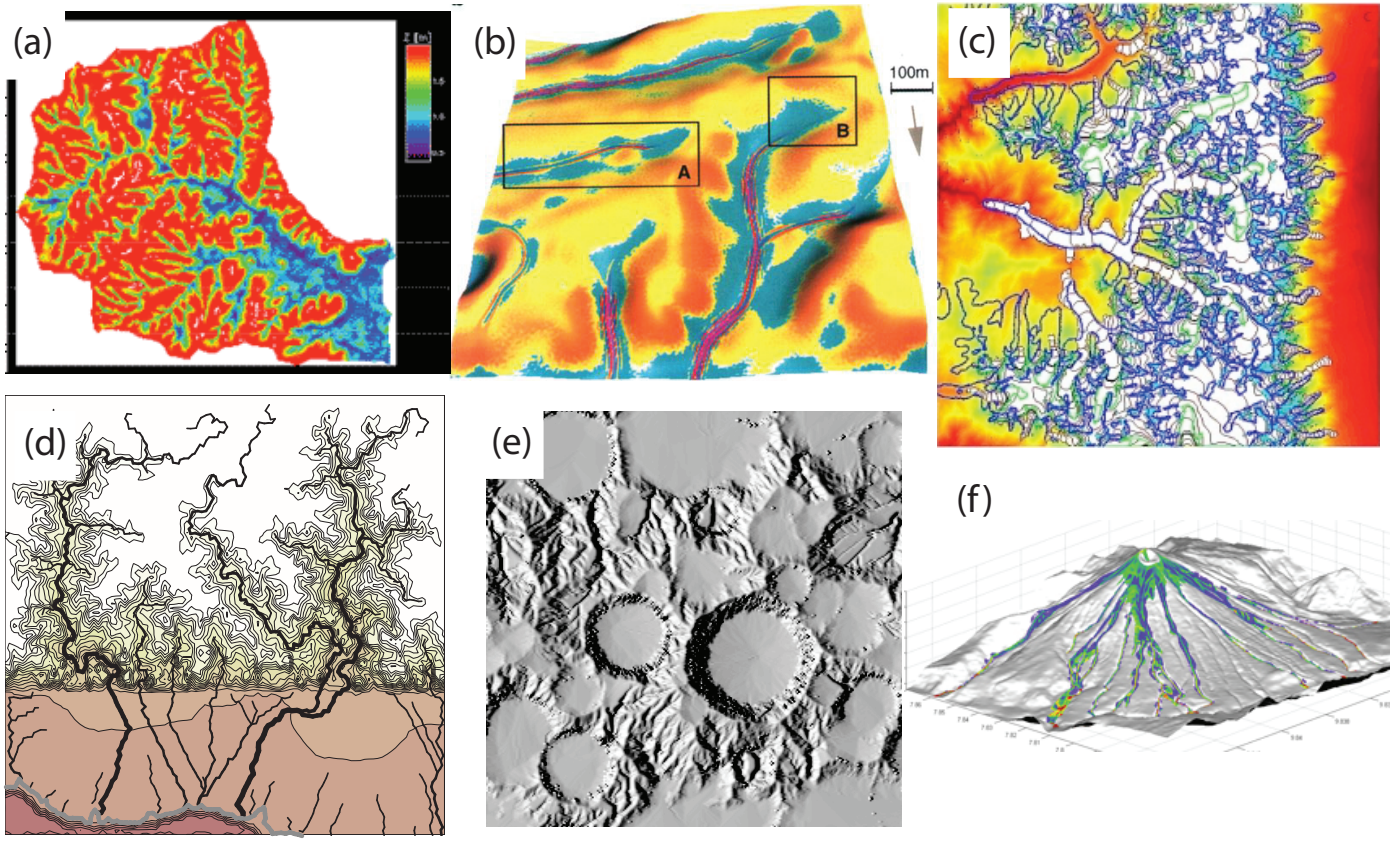
SIBERIA (Willgoose and Riley, 1998), J. Prancevic (unpubl.), CHILD (Tucker et al., 2001), SIGNUM (Refice et al. 2012), Simpson & Schlunegger 2003, Braun & Sambridge 1997, mARM4D (Cohen et al., 2013), Castelltort et al. 2012, Tellus (CSIRO)

Many commonalities in earth-surface process models

CATCHMENT HYDROLOGY
(Ivanov et al., 2004)

SOIL EROSION
(Mitas and Mitasova, 1998)

GLACIER DYNAMICS
(Kessler et al., 2006)



LANDSCAPE EVOLUTION
(Tucker and Hancock, 2010)

IMPACT CRATERING AND DEGRADATION
(Howard, 2007)

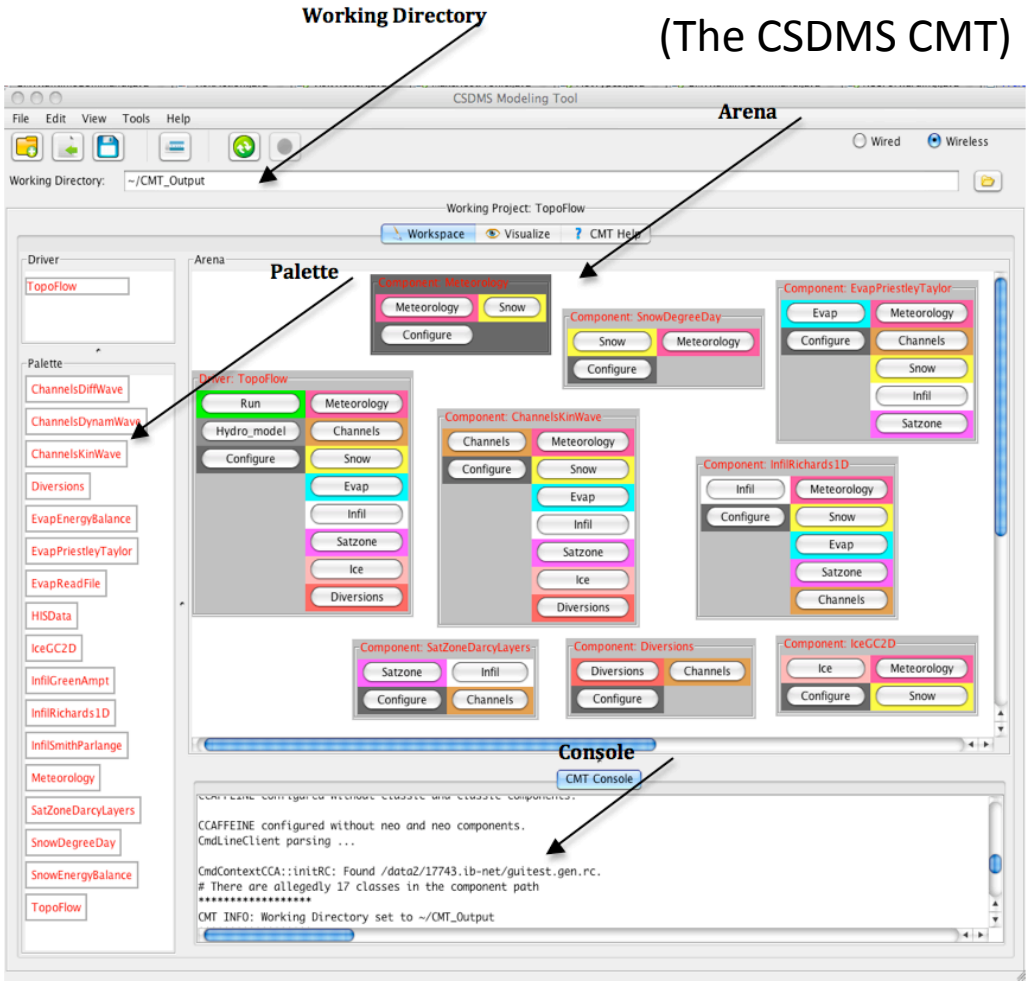
LAVA FLOWS
(Kelfoun et al., 2009)

Of those 9 models:

- 3 are open access through CSDMS (SIBERIA, CHILD, SIGNUM)
- 1 is CMT compatible (CHILD)

CSDMS
COMMUNITY SURFACE DYNAMICS MODELING SYSTEM

(The CSDMS CMT)

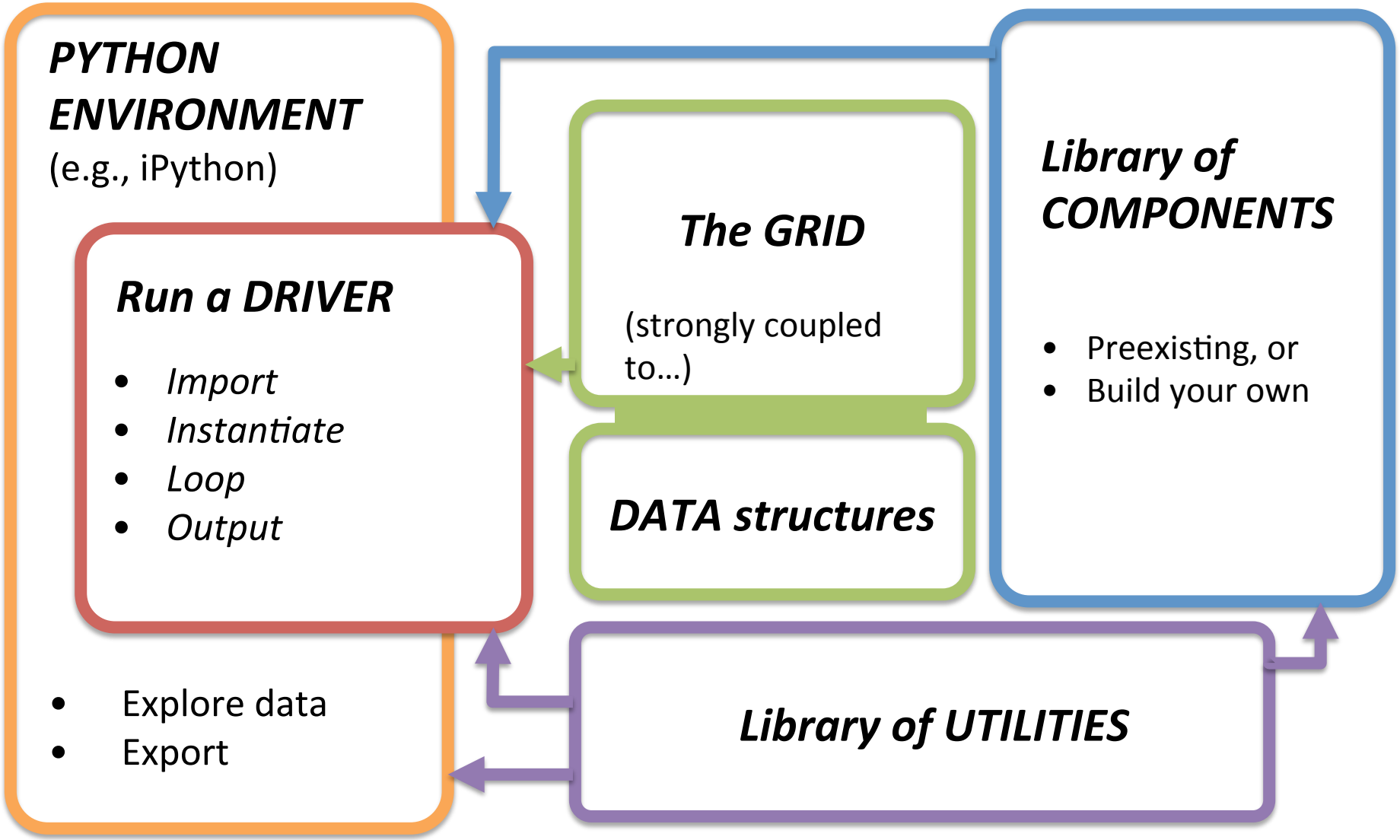


Our objectives:

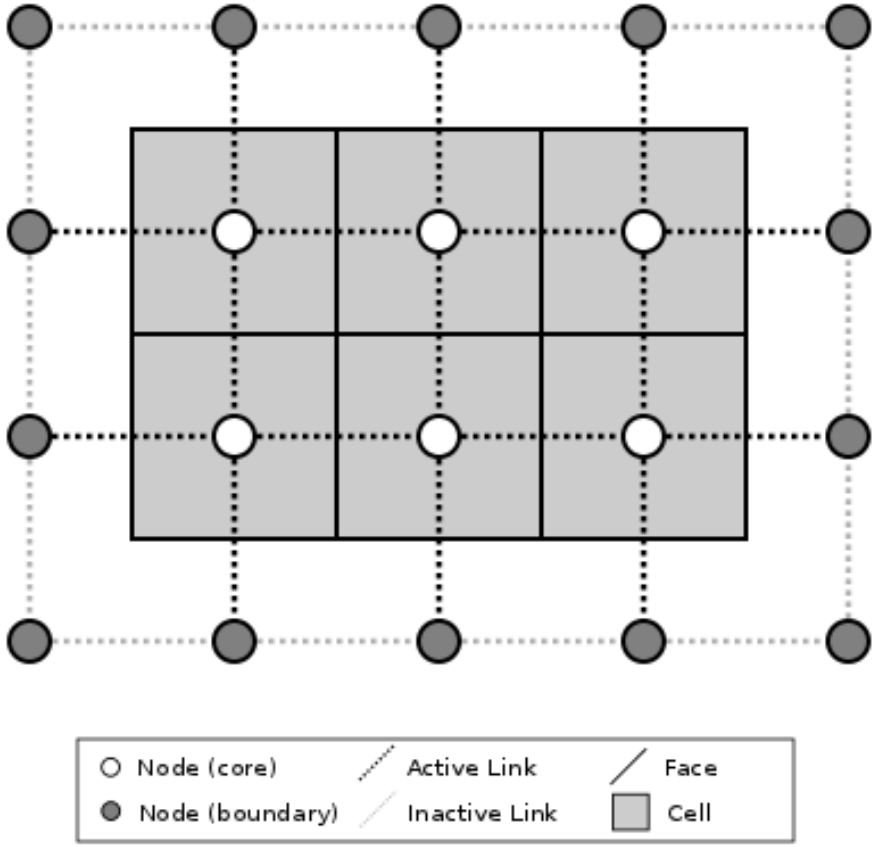
- Open-access
- Non-proprietary
- CSDMS CMT compatible, from the ground-up
- Quick to learn
- Quick to code
- Modular
- Flexible

python

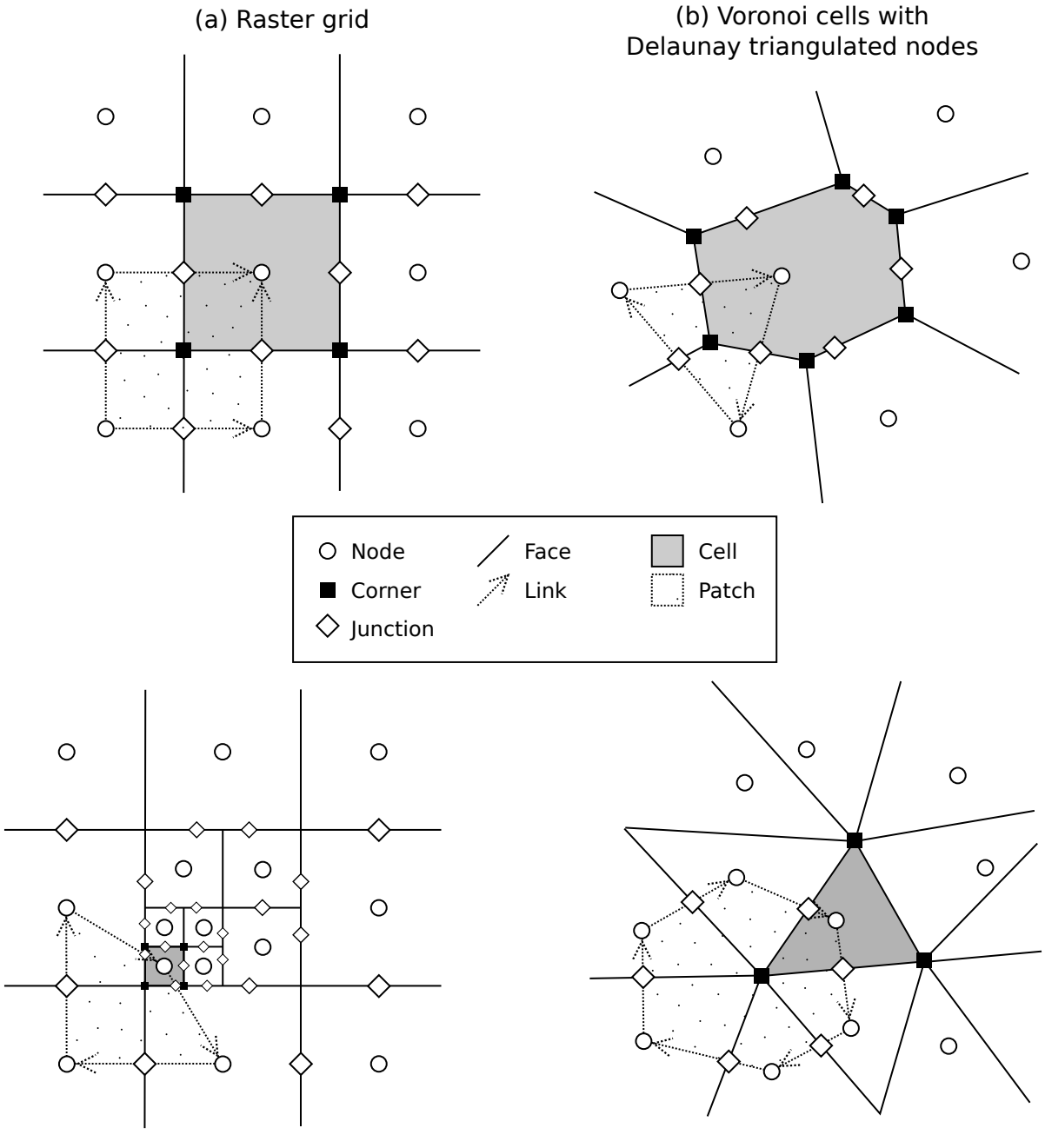
Architecture



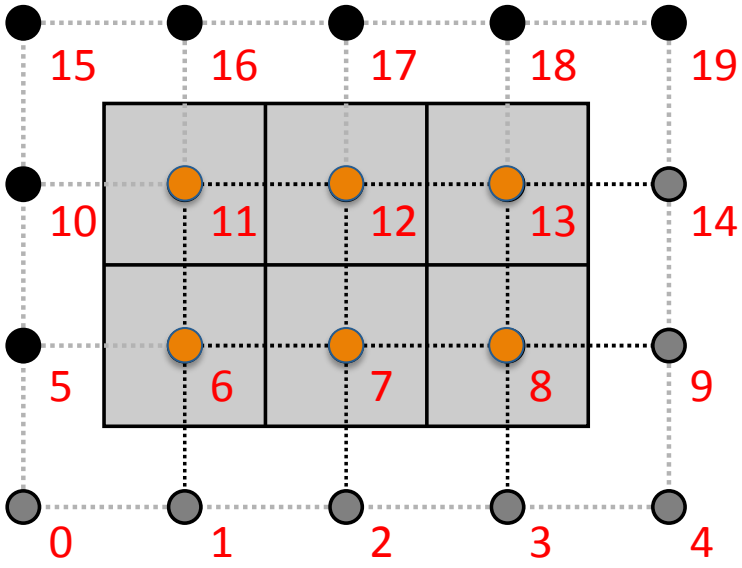
The grid



- CHILD-like
- Data defined on *nodes* or *links*
- Explicit control of *cell status*
- Functions describe geometry
- Not limited to raster…


(a) Raster grid
(b) Voronoi cells with
Delaunay triangulated nodes
Node
Corner
Junction
Face
Link
Cell
Patch
(c) Quadtree grid
(not yet implemented!)
(d) Delaunay triangle cells with
Voronoi polygon patches

The data



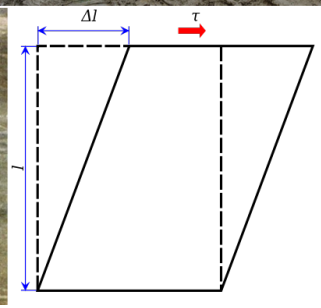
- *Nodes* or *links*
- 1D arrays of values
- Numpy
- Grid and data are wrapped together in a single object

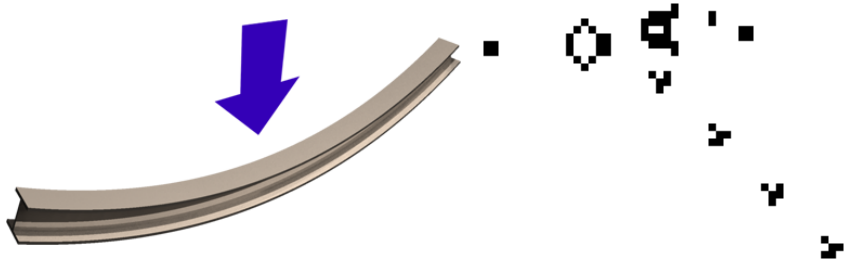
core_nodes: [6, 7, 8, 11, 12, 13]

surface_elev[core_nodes]: [2., 2., 2., 3., 3., 3.]

The components

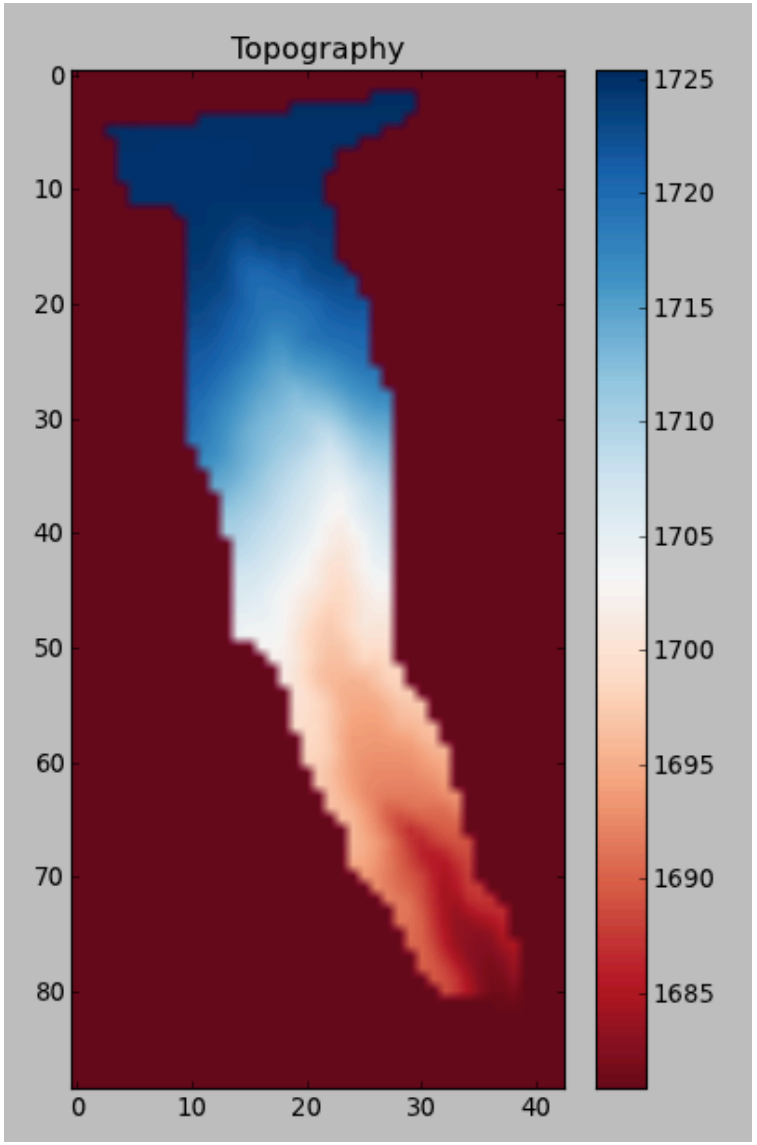
- Describe individual surface processes
- “Plug & Play”
- Standard interface
- Use the library, or BYO



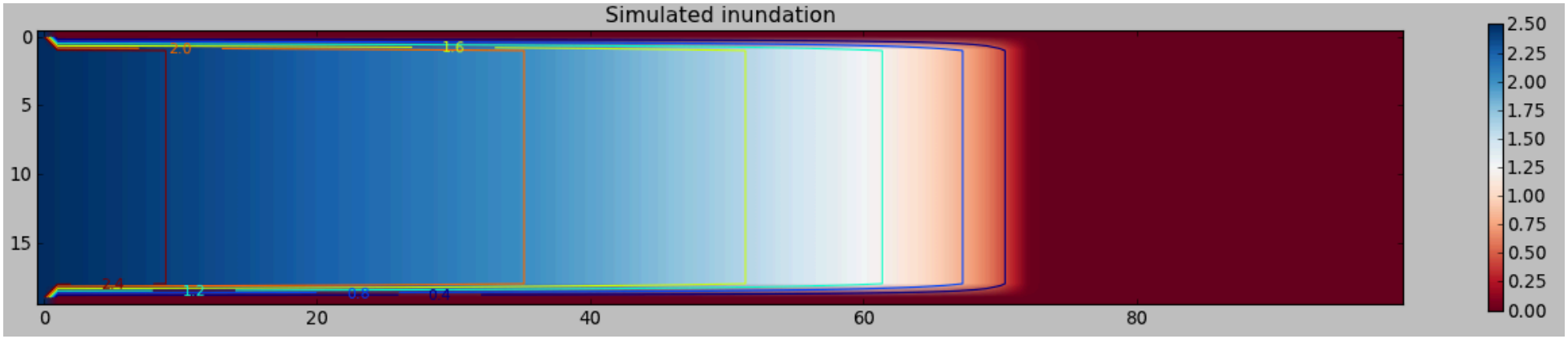


The utilities

- Read DEMs
- Read .txt
- Output routines (e.g., netCDF)
- Visualization tools
- Other repetitive tasks

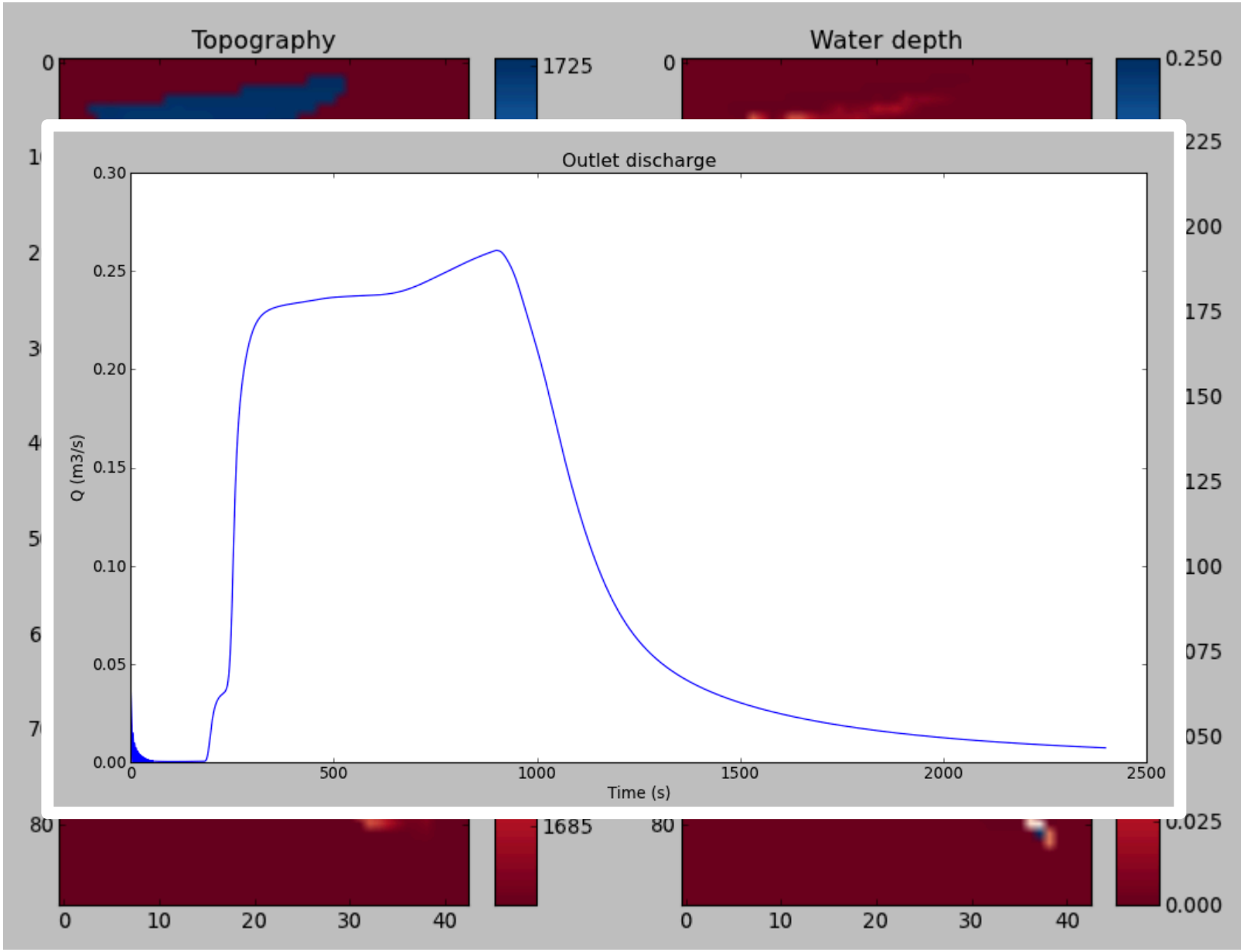


Flood-wave propagation

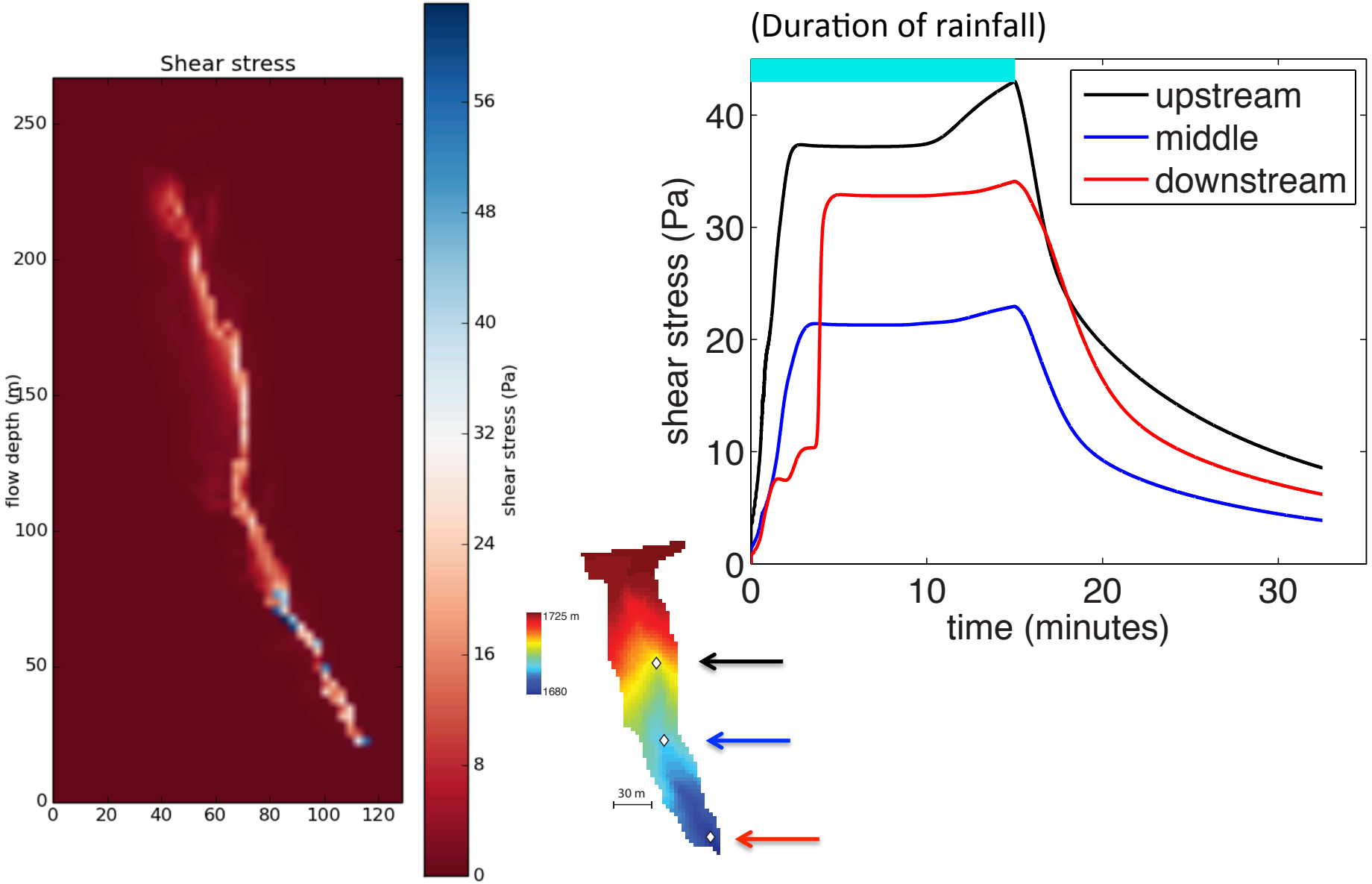


- Follows math laid out in Bates et al. (2010)
- Calculates flux divergences at nodes
- Very simple to implement in driver

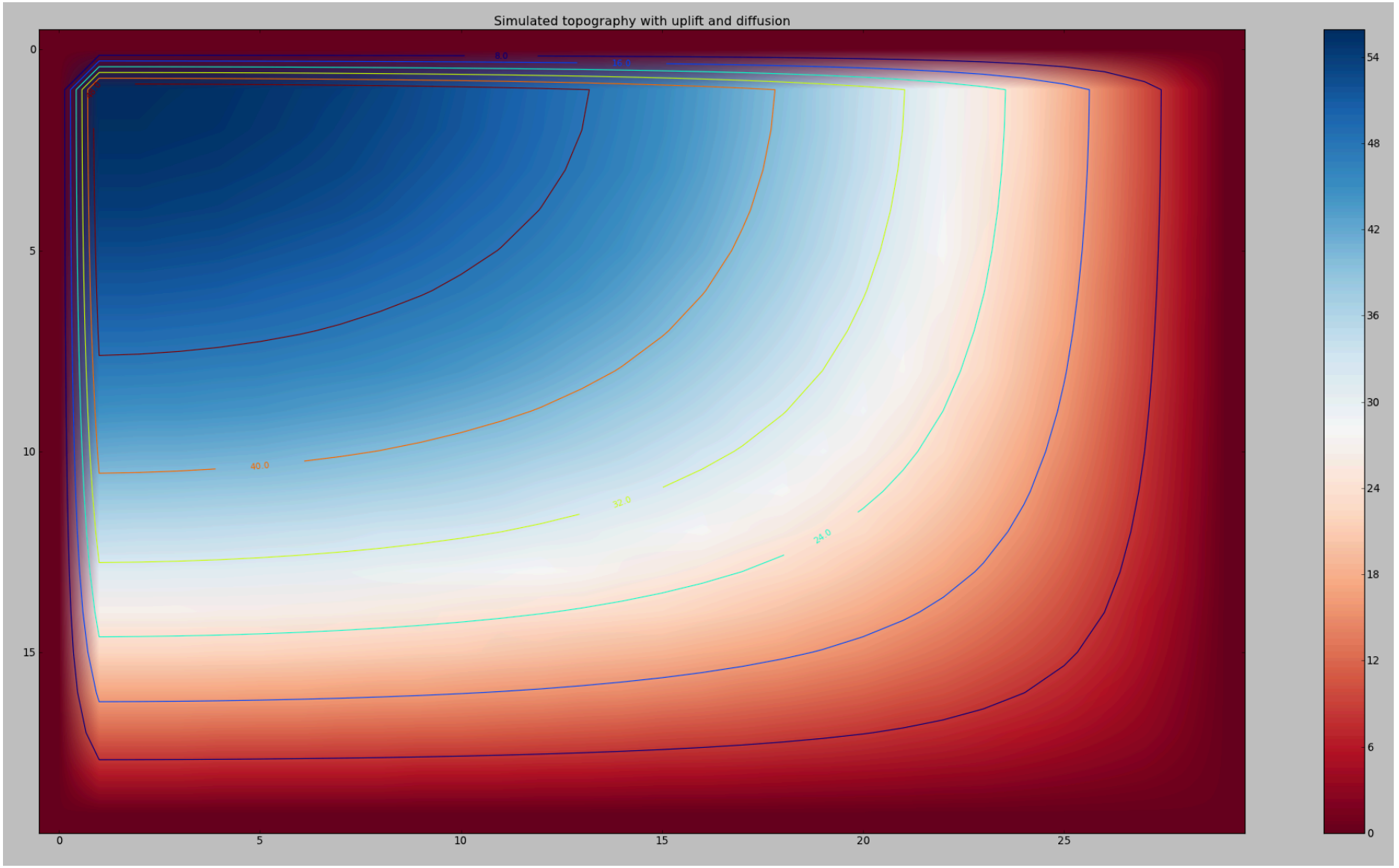
Overland flow on a DEM



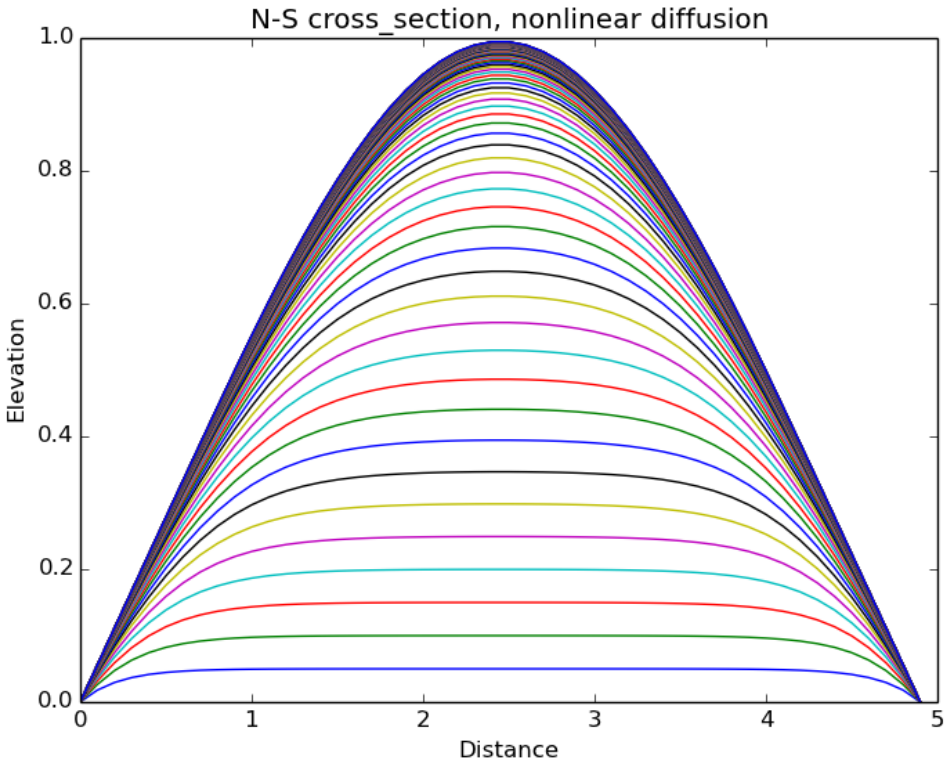
Couple shear stress component…

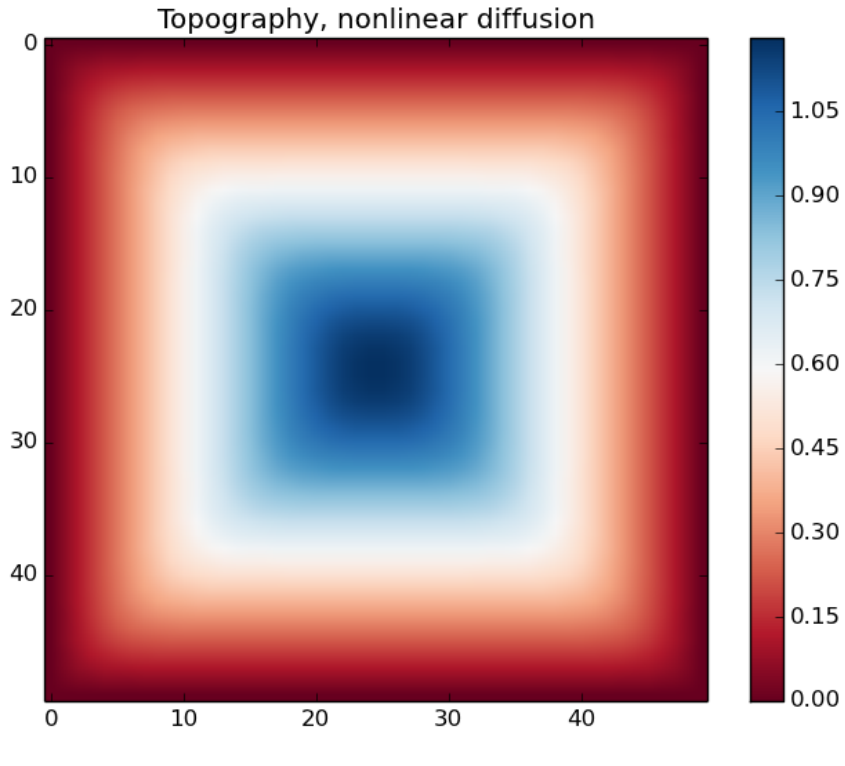


Hillslope diffusion & tectonic uplift



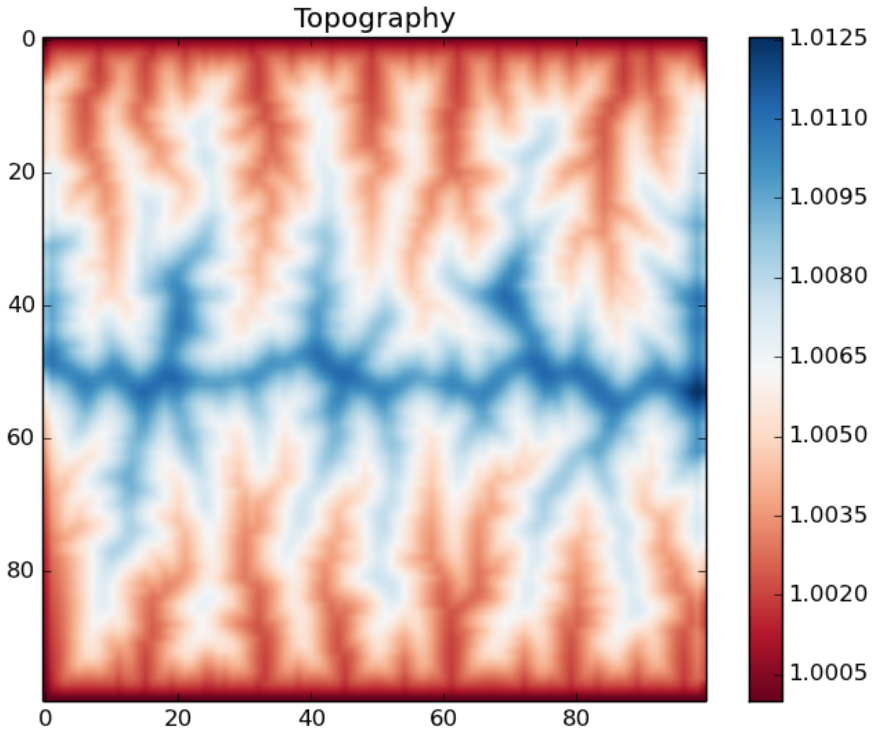
Nonlinear diffusion

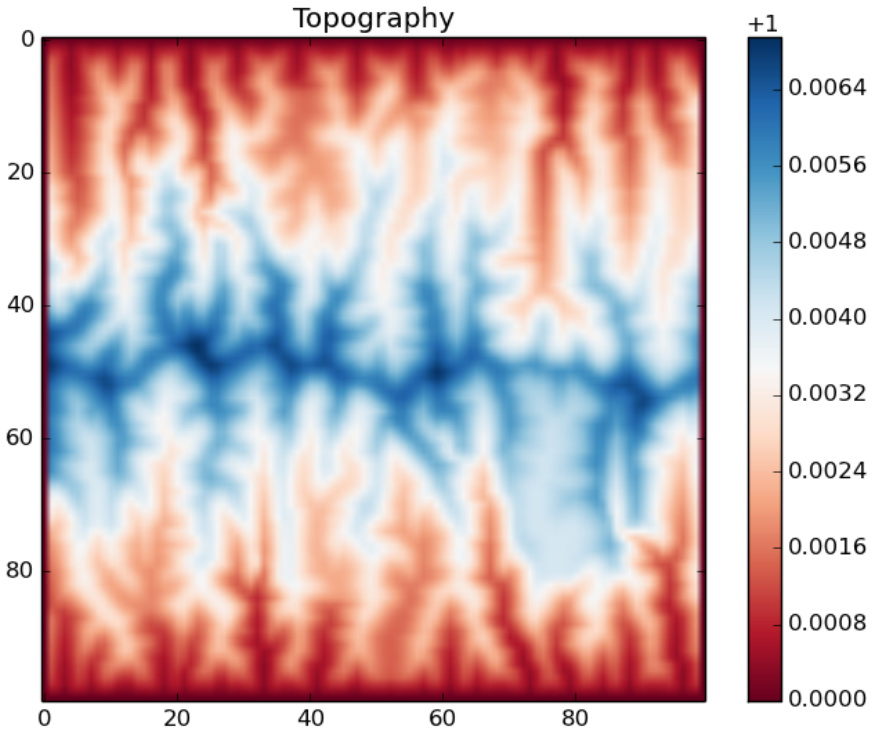




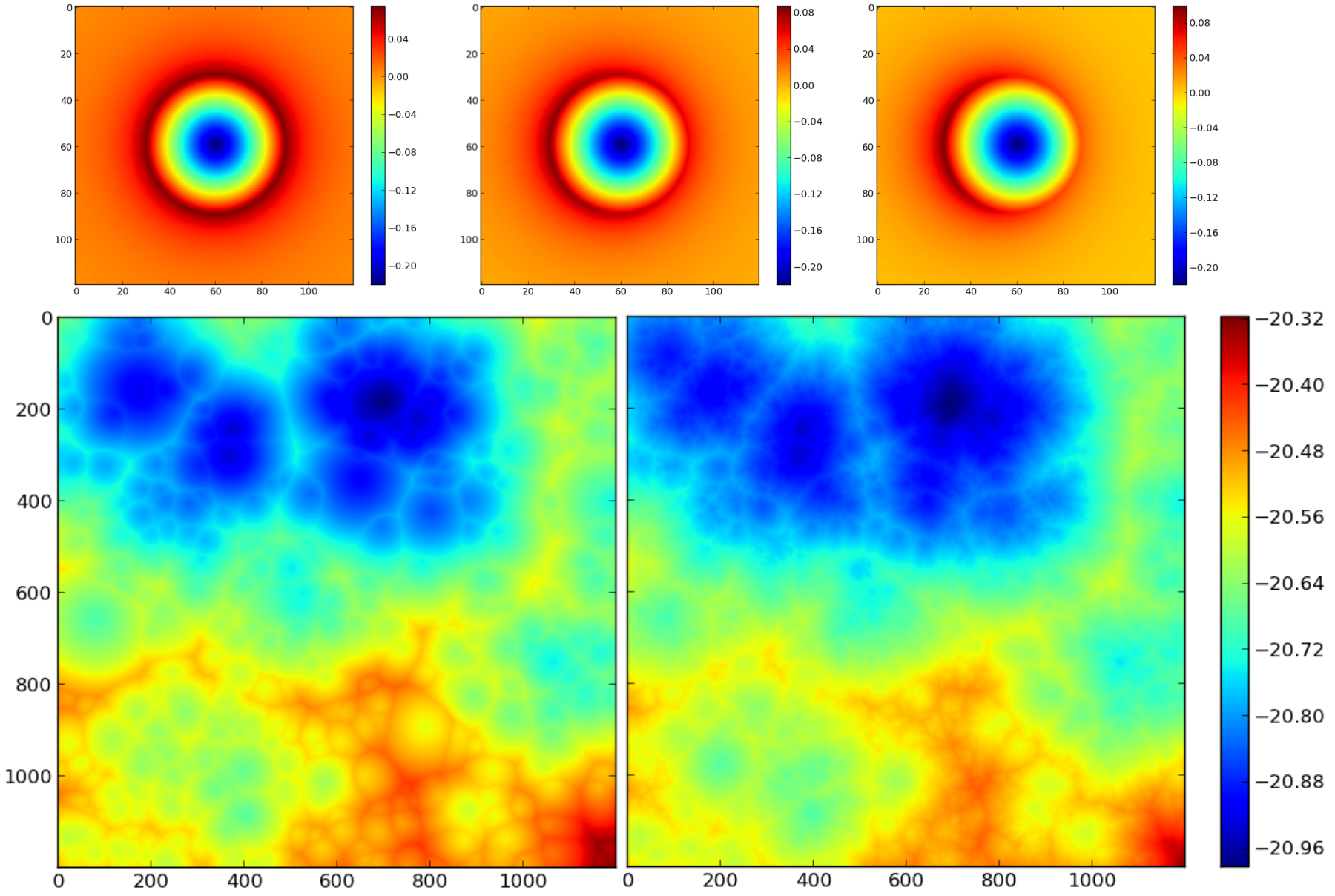
Stream power

Hillslope-channel coupling

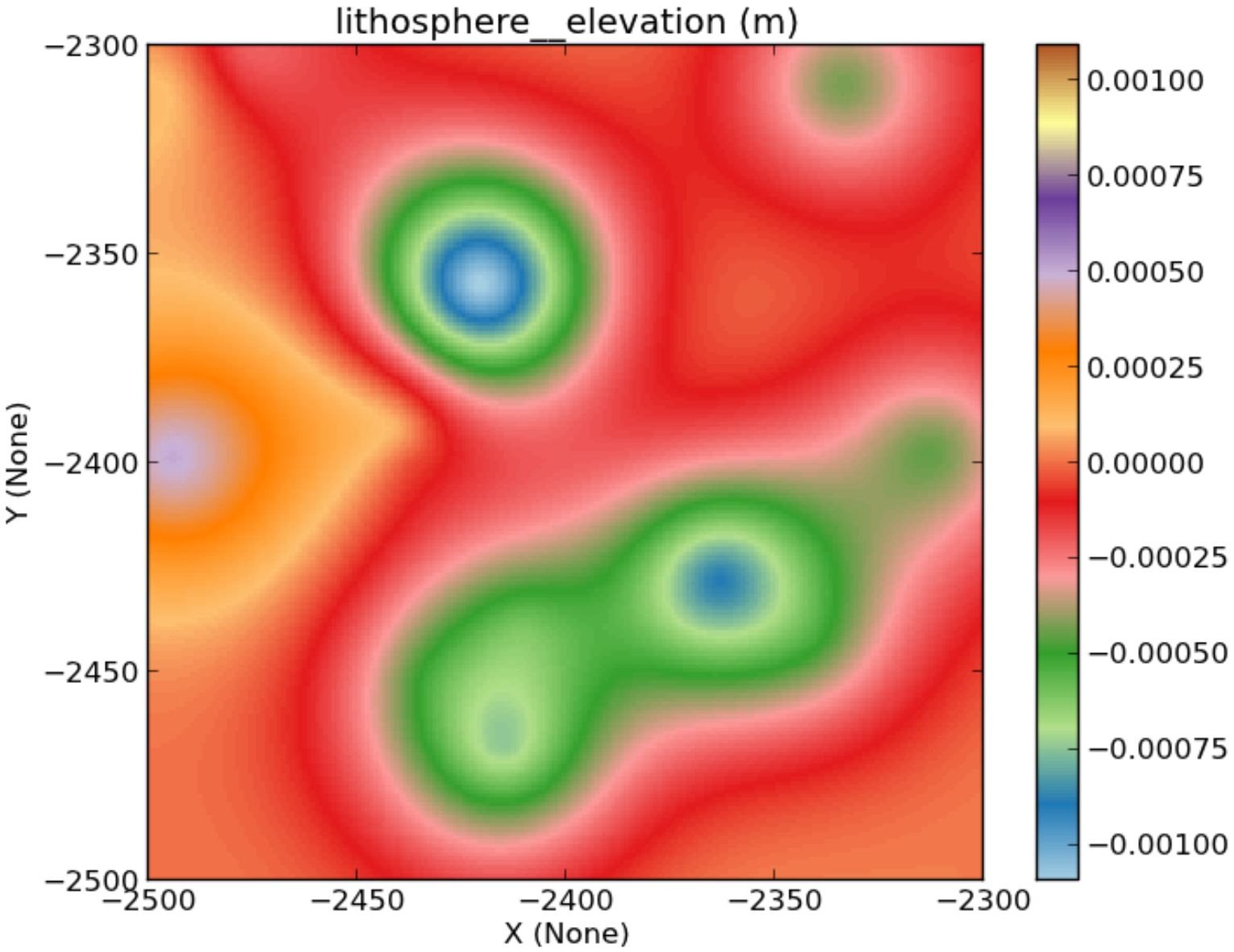




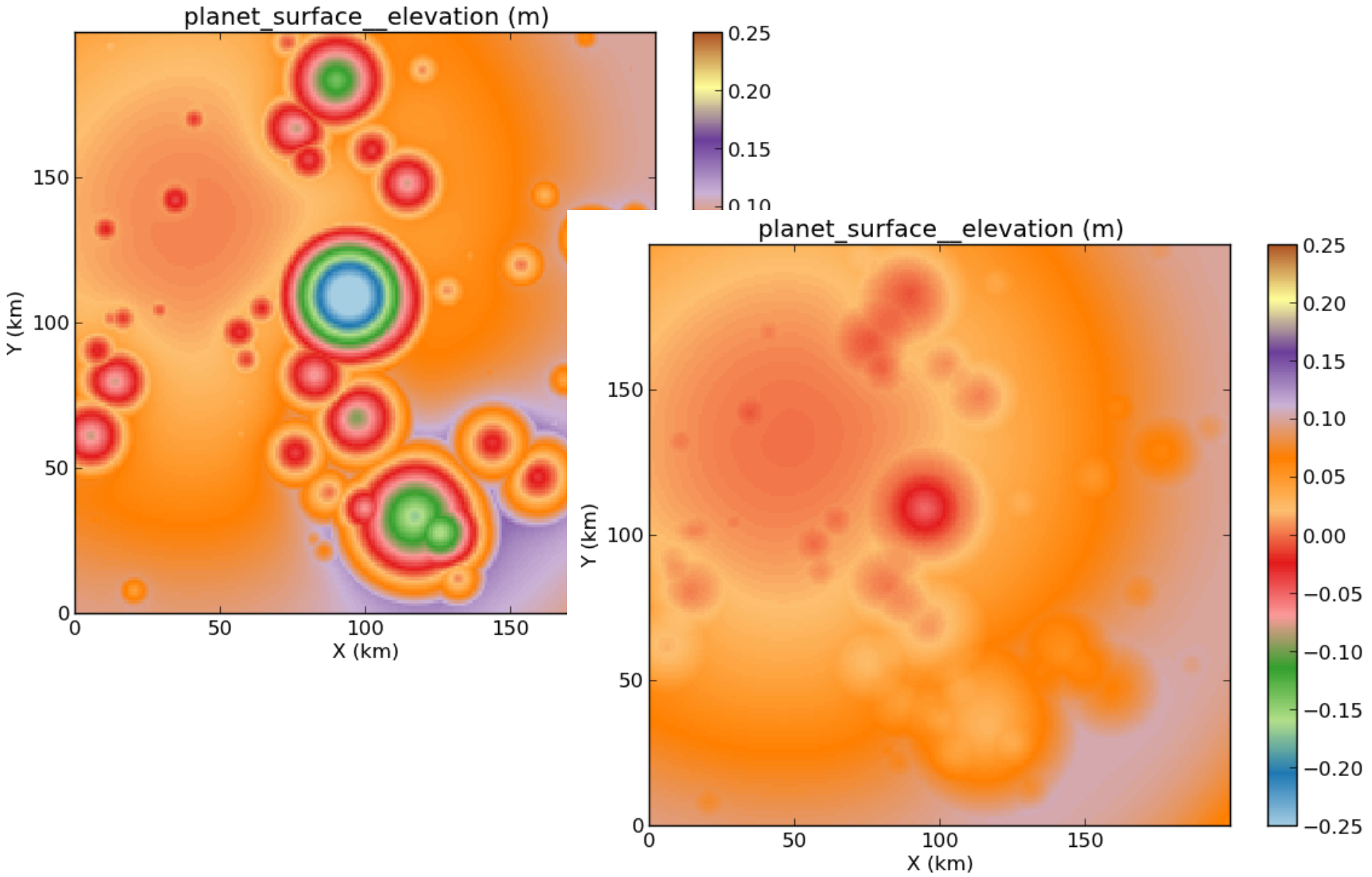
Impact cratering



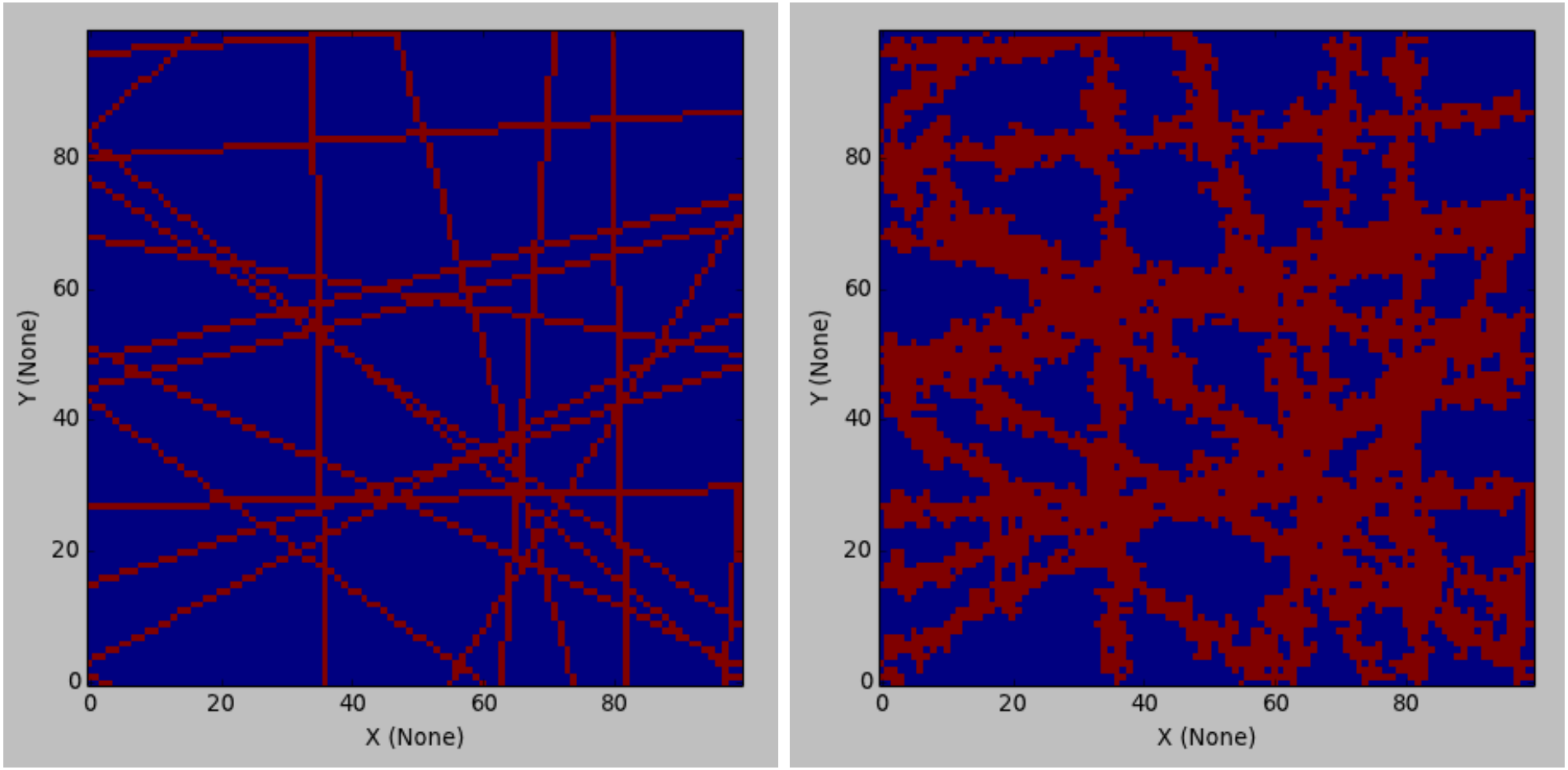
Flexure


lithosphere__elevation (m)
−2300
−2350
−2400
−2450
−2500
Y (None)
−2500
−2450
−2400
−2350
−2300
X (None)
0.00100
0.00075
0.00050
0.00025
0.00000
−0.00025
−0.00050
−0.00075
−0.00100

Coupled flexure



Cellular automata



Simple CA model of weathering around fractures (red) into a medium (blue).

Summary

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

Session plan

Technical aspects

- “Developer” vs. “User”
- Work with the source code
- Instructions on web:
  landlab.readthedocs.org

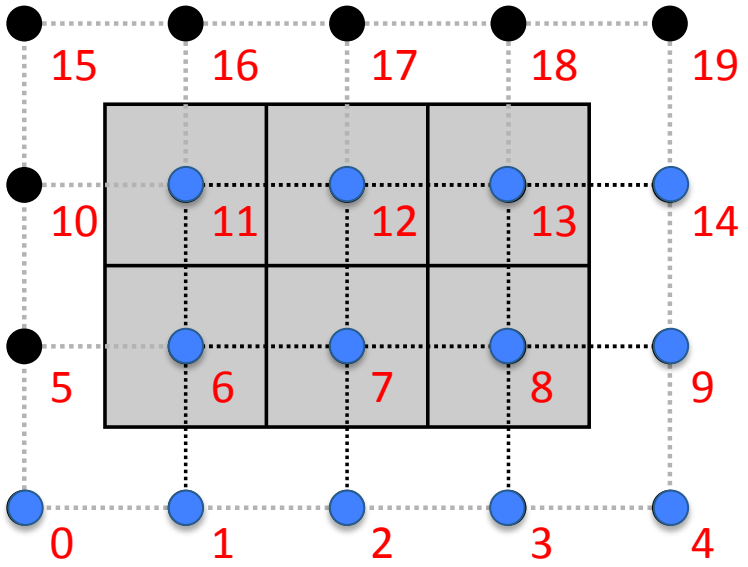
From your directory, in a terminal:

$ python setup.py develop

Start python/ipython, &

>>> import landlab [as ll]

- Landlab is *object oriented*
- The grid and components are all objects



- Import -> instantiate -> loop -> output

- Data lives "inside" the grid object, as numpy arrays:

```
>>> mygrid = ll.RasterModelGrid(num_rows=4,
                                num_cols=5)
>>> mygrid.create_node_array_zeros(
                        'planet_surface__elevation')
>>> mygrid.at_node('planet_surface__elevation')
```

- Grid gets passed around between components
- (see examples)

How fast is it?

- Fast to code
- Exploit Numpy routines
- Vectorized
- Cython
- PyPy…?

- …”Faster than you’d think”