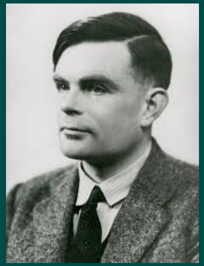
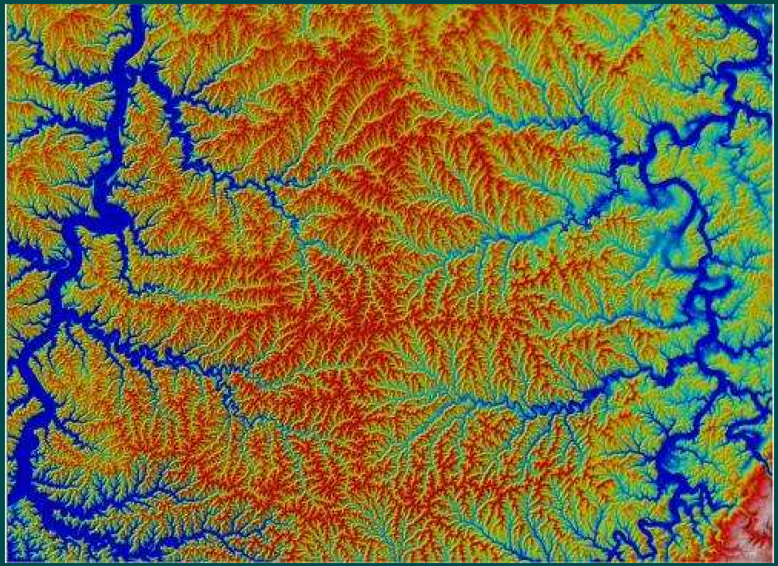
Pattern formation - morphogenesis

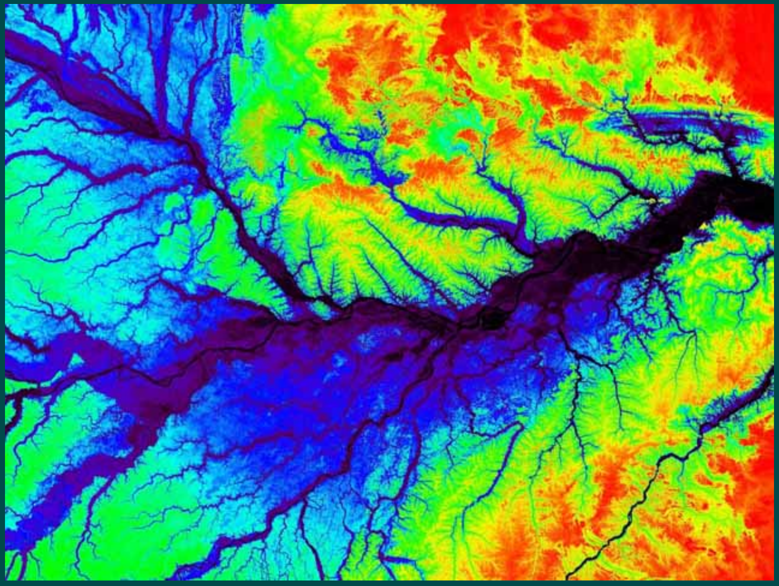


- All zebras are different
- Is there a recipe for each zebra?
- If its the same recipe, why are they different
- Emergent patterns – statistical variations

- Geomorphology:
- How are landscapes formed?
- (Sierra Nevada, Ca)

Specifically: pattern formation by erosion



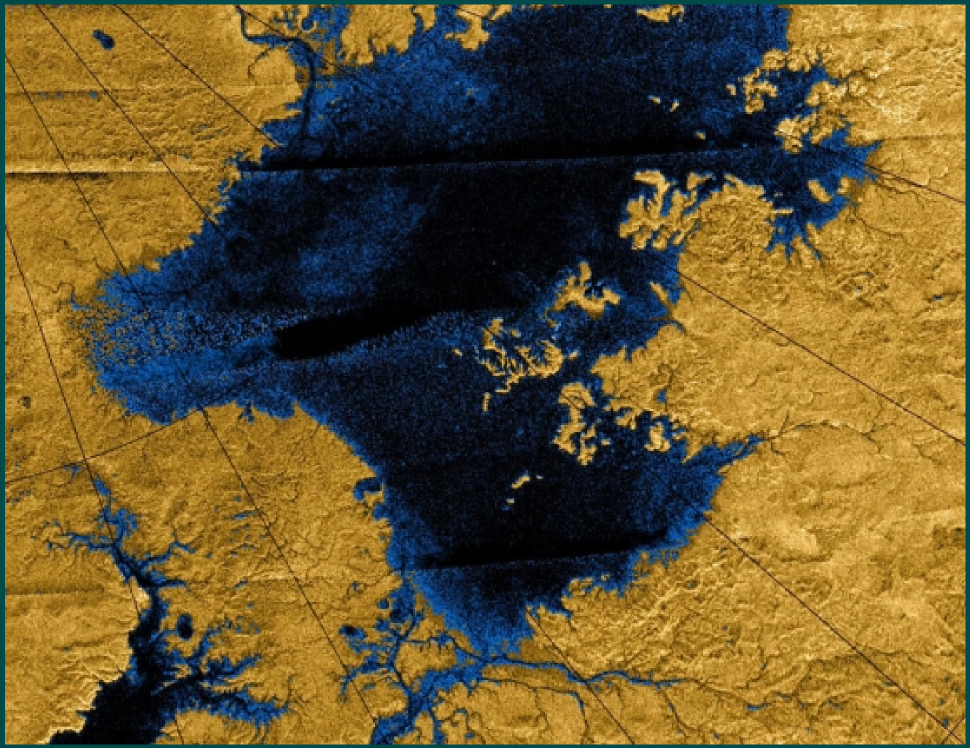


River networks are complex evolving systems.

Crucially important to our planet's health

Can we begin to understand how they develop

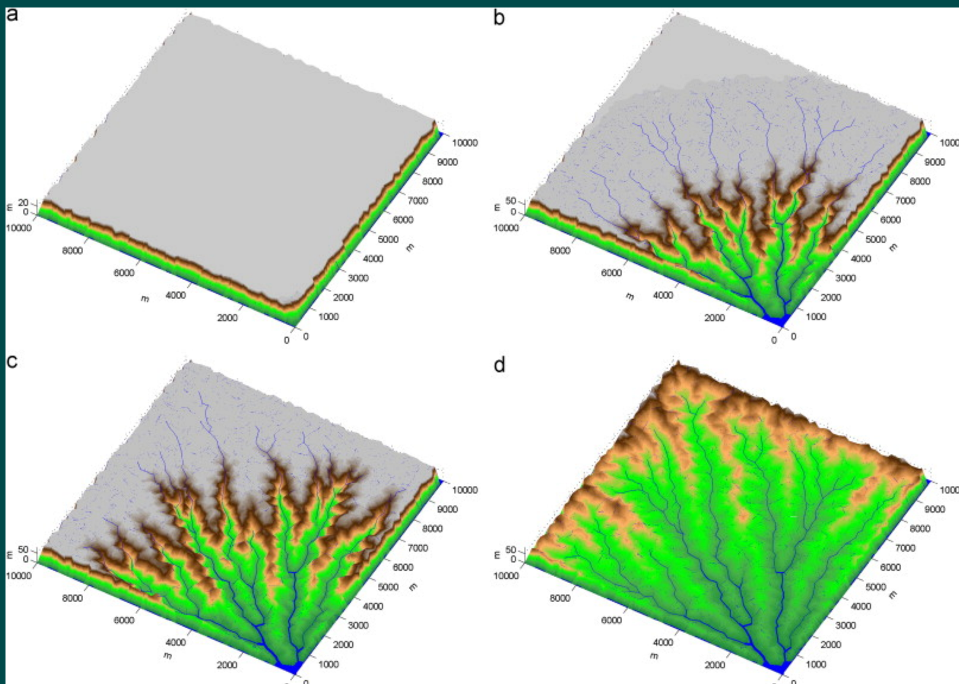
Not just our planet

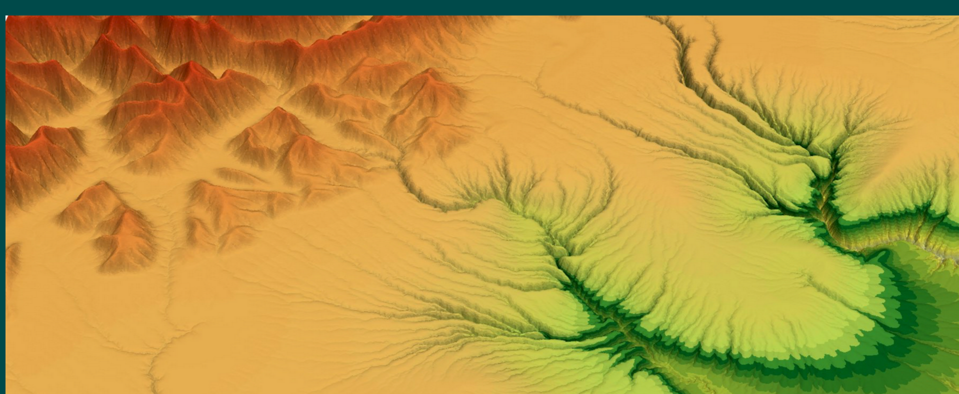


Images from Cassini mission show rivers draining into a lake on Titan (a moon of Saturn)

http://news.mit.edu/2012/river-networks-on-titan-0720

Models of river network evolution





How do recognizable patterns emerge from random noise?

What sets the scales?

How do the scales depend on the parameters in the model

Key elements of landscape erosion models

Topographical map *h*(*x*,*y*,*t*) (images are snapshots of *h*)
h is a dynamical variable: evolves in time – how?

Continuity: $\partial_t h + \nabla \cdot q = 0$ *q* is the sediment flux

Fluxes have convective and diffusive components:

$$q = uh - D\nabla h$$

Convective flux from deterministic movement of material
Diffusive flux from random spreading (e.g. by wind)
Pattern formation requires <u>feedback</u> in the convective flux

What is the feedback mechanism?

Coupling of flow and topography

Steeper slopes have more flow – more flow leads to more erosion and even steeper slopes – and so on

A typical landscape model:

$$\partial_t h = D \nabla^2 h - K A^{1/2} |\nabla h| + U$$

K is a constant (stream power coefficient)

A is the area of the basin draining to that point

U is the upwelling (constant velocity)

A simpler model

- Landscape evolution models produce the networks shown in earlier slides
- But a bit too complicated for one afternoon
- Program a simpler model with similar qualitative features

A stochastic river model

Classical landscape models are deterministic

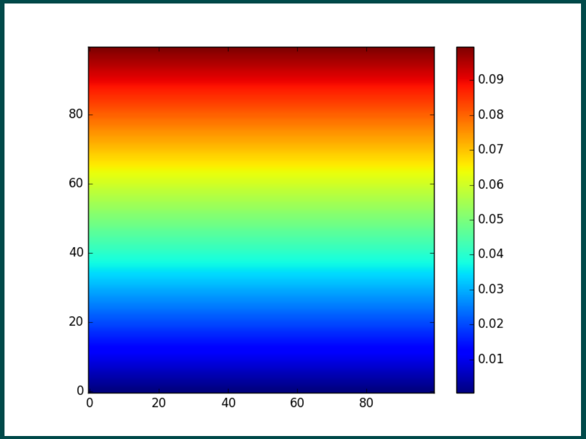
Randomness comes only from initial conditions

Here we have a smooth initial condition but with randomness in the dynamics

Elevation map *h*(*x*,*y*,*t*)

At *t* = 0 it is a flat slope with a gradient *I*:

$$h(x,y,t=0)=Iy+10^{-6}R$$



Periodic boundaries in *x*, river (sink) at *y* = 0.

Add random rainfall

Create empty list for wet sites

Pick a site (x, y) at random and add to list

Now we need to decide which of the 4 nearest neighbors it goes to next (rain only moves at right angles in our model)

Add chosen site to list

Rain does not go up but we need a way to choose among different down sites

Where does the water go

Our map *h* is on a coarse scale

Small scale variations in topography may mean that the water does not always flow directly down (i.e. steepest descent) but meanders

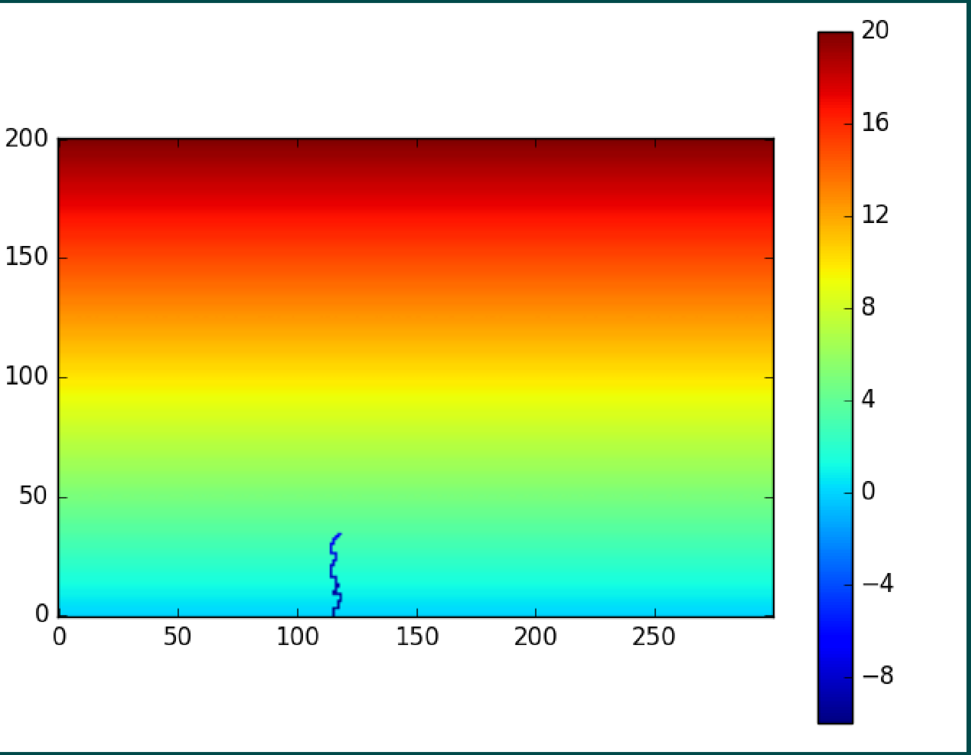
Choose from candidate sites with a weight that is a function of the height difference. Site *i* has *n* neighbors (n = 4) and the probability to select site *j* should be:

$$P_{ij}=\frac{w\left(h_i-h_j\right)}{\sum_{k=1}^{n} w\left(h_i-h_k\right)}$$

$$w(x)=e^{\beta x} \quad \text{if } x\geq 0$$

$$w(x)=0 \quad \text{if } x<0$$

Erosion step



Once the water reaches the stream (y = 0):

Erode all the places it visited by a constant Δh

Repeat for next rain drop

Details in Leheny1993.pdf Figure 1 – PRL 71: 1470

Simulation parameters:

300 x 200 grid

l = 1.0

Δh = 10

β = 0.05

r = 200

100000 drops

takes about 5 mins on my laptop

I suggest smaller shorter simulations to begin with

Avalanches

Landscapes develop with (unrealistically) large gradients – rivers are thin and do not grow in width

Add avalanches: reduce extreme height differences between each x,y and neighbors x',y' to below some threshold *R*

$$\text{if } \max \Delta h = h(x,y) - h(x',y') > R: \quad h(x,y) = h(x,y) - 0.25\,\Delta h$$

Repeat until all $\Delta h < R$

Tasks

- Write a program to evolve the height of an initially flat slope

  Make a correctly oriented plot of the height map
- Add code to detect avalanches
- Write a program to detect the rivers

  Make a plot to show the river networks
- 1/3 points for each of the above

- Extra credit (0.2) Make a 3D plot of the topography with the river network superposed on top

Computational hints

- The crucial step in the topography evolution is to figure out how to select 1 of the 4 possible outcomes with the correct weight
- The random walk may revisit the same point more than once – but we only want to erode once

  list(set(wet)) will generate a unique list of entries
- Small random noise in initial state will limit repeats
- Start small and print things to allow you to follow what is happening. Or use a debugger (better)

Finding the rivers

This is slow – so only do it occasionally (for the plots)

Initialize array: r = np.ones(nx,ny)

From each grid cell find the steepest descent path to the sink. Add 1 to r at each cell it visits.

Threshold r by some parameter R ~ 100 to make a binary image