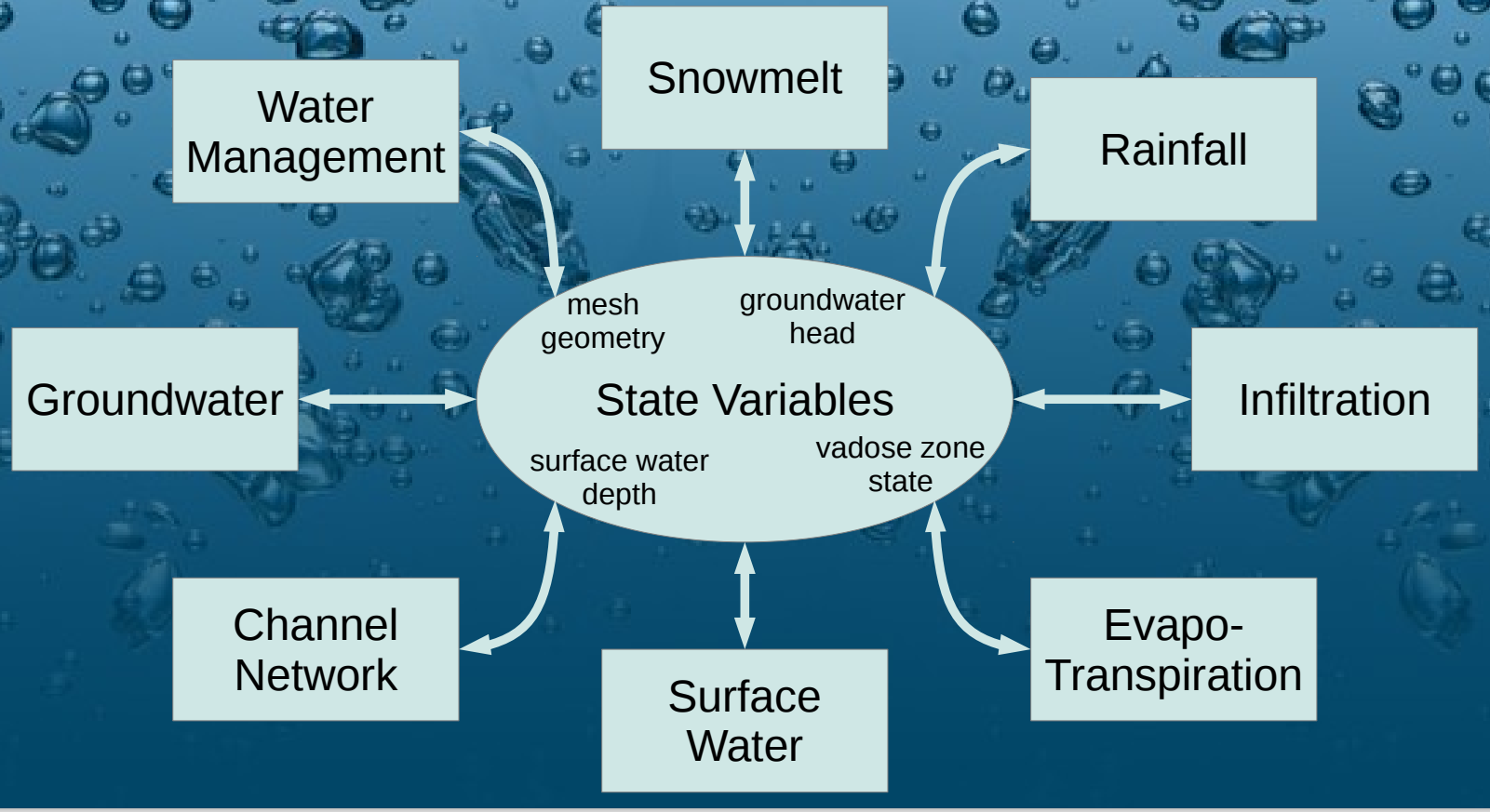
# ADHydro Simulation



# Recent Progress

Improved Talbot-Ogden infiltration model

- Model improvements and re-validation
- Performance improvements

Integrated surface water, infiltration, and groundwater into main program

- Complexities integrating infiltration and groundwater
- Tested on open book watershed

Output state variables in HDF5 format for visualization in Paraview

# Recent Progress

Uploaded to Alfresco instructions and python scripts for generating simulation mesh input files

Integrated Noah-MP land surface model in to ADHydro evapo-transpiration model

- Noah-MP Fortran code called by ADHydro C code

Developed channel network model

- Program for generating channel network topology input files
- Channel network simulation model

# Integrating Infiltration and Groundwater

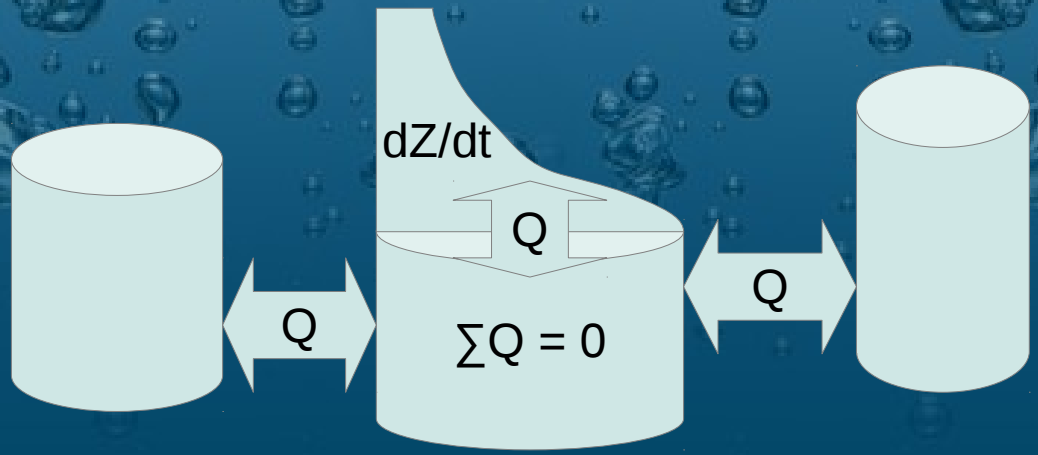
The groundwater model acts as though the mesh is a series of open-topped buckets. Any amount of water can be taken from the top of one bucket and added to another.

# Integrating Infiltration and Groundwater

The infiltration model has equations for the rate at which the groundwater front rises and falls based on groundwater head. Water cannot be taken from or added to the top of a bucket at a rate different than this.

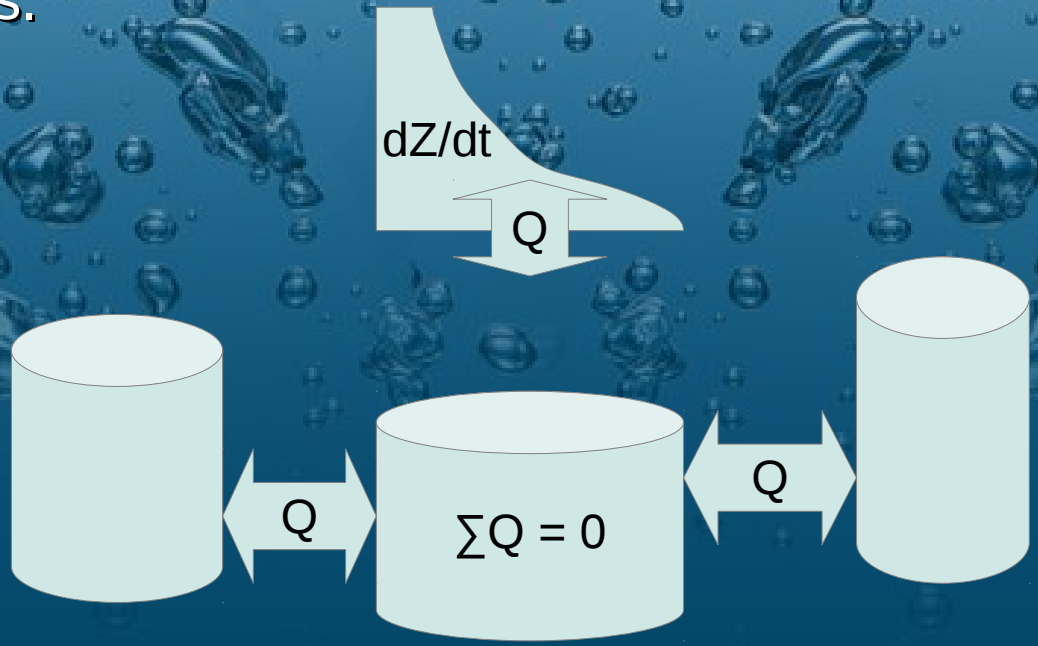
# Integrating Infiltration and Groundwater

In reality, the pressure head changes at the speed of sound to make the flows equal. Must decouple pressure head from groundwater front height.



# Integrating Infiltration and Groundwater

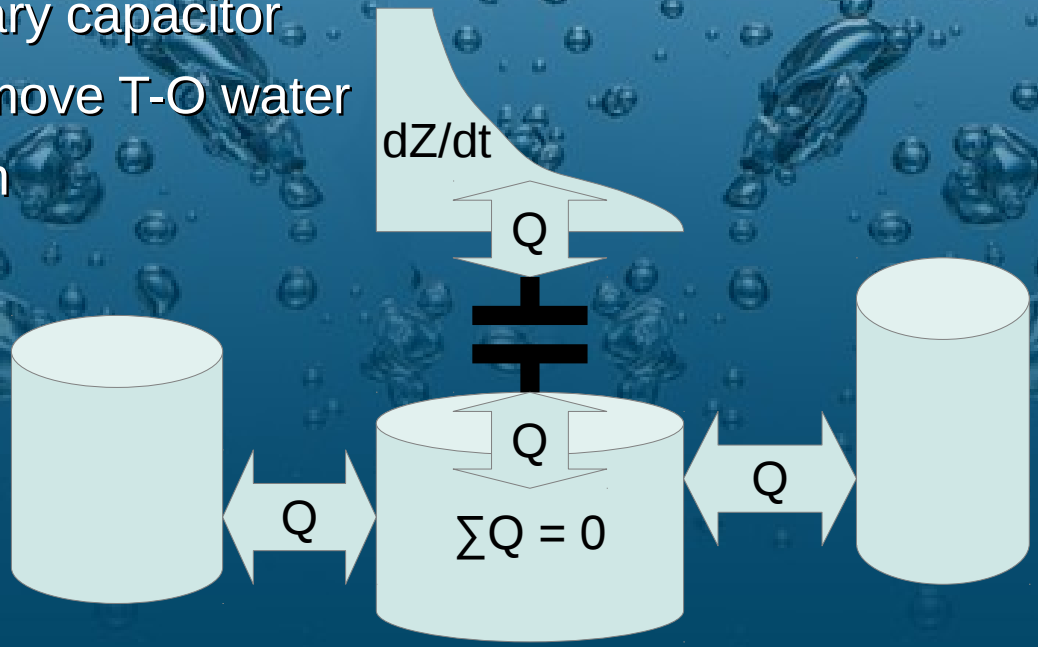
We want to run infiltration and groundwater as two separate loosely coupled models.



# Integrating Infiltration and Groundwater

Possible solutions:

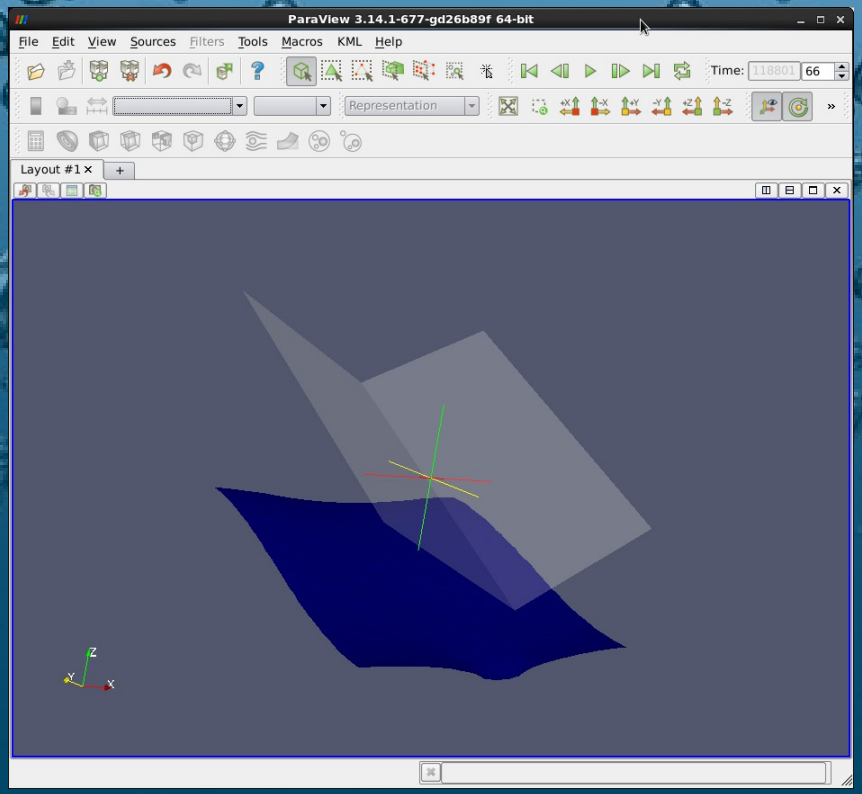
- Imaginary capacitor
- Add/remove T-O water
- Iteration



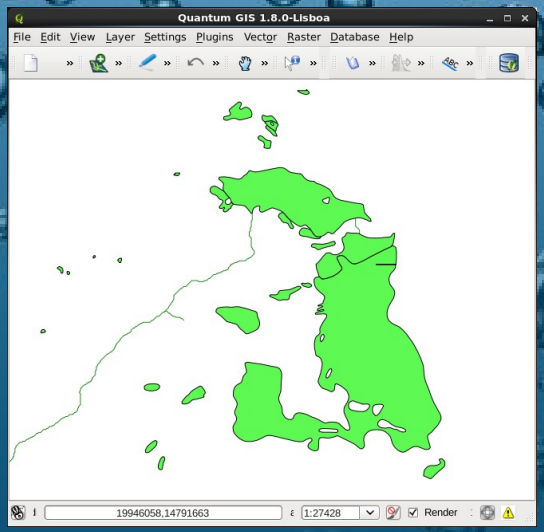
CI-WATER

A Utah-Wyoming Cyberinfrastructure
Water Modeling Collaboration

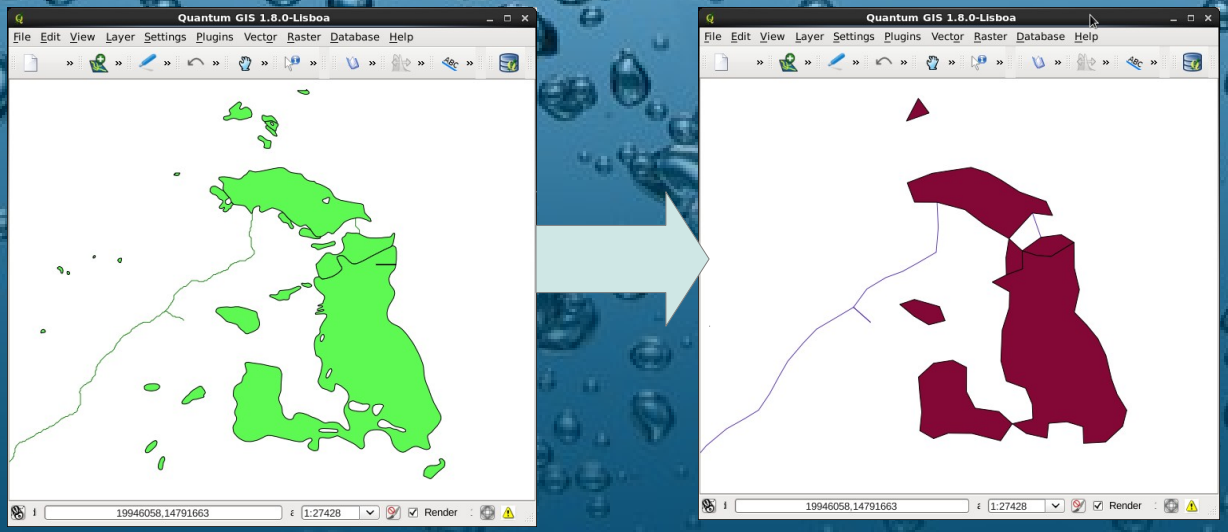
# Visualization With Paraview



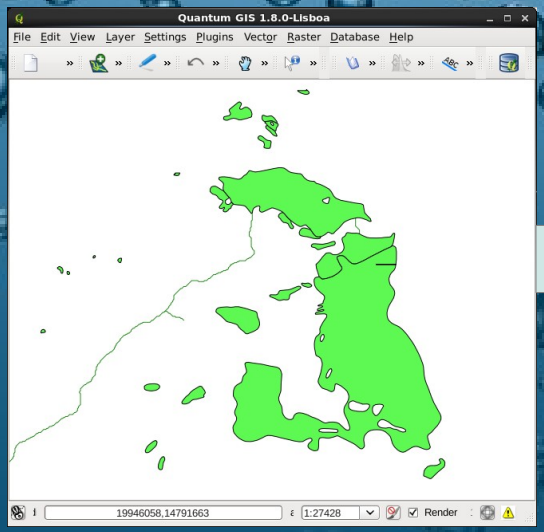
# Channel Network

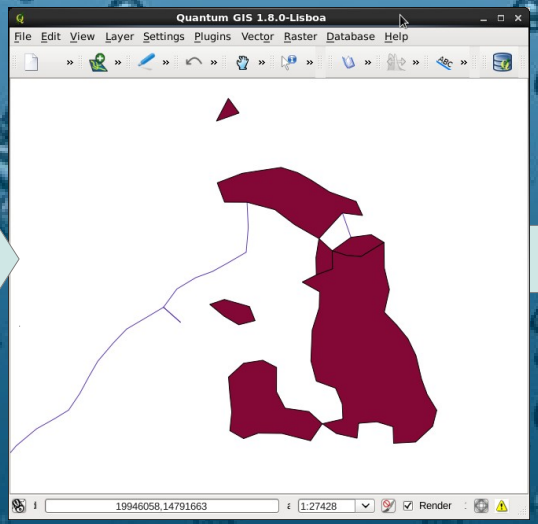


# Channel Network

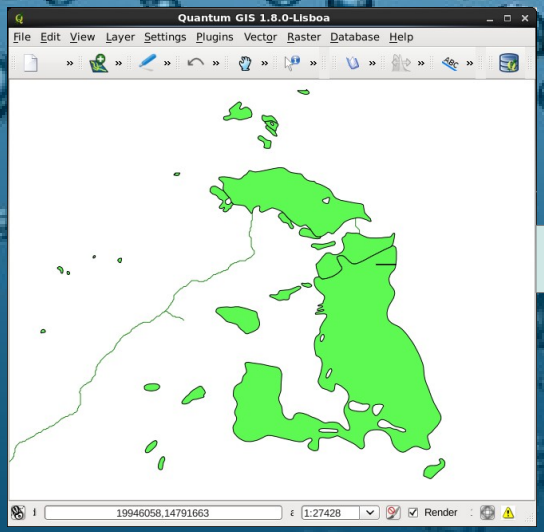


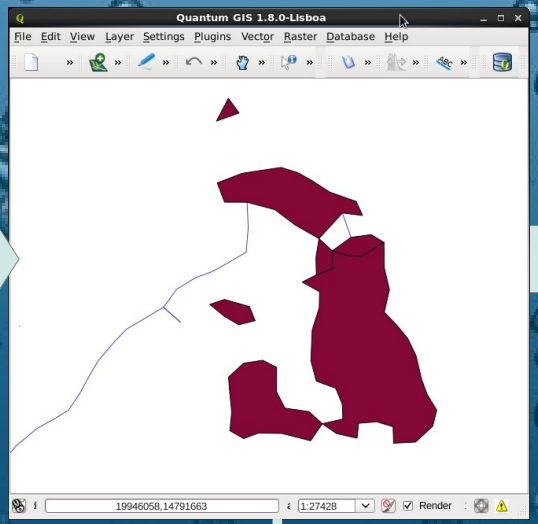
# Channel Network




# Channel Network





```
Struct {
    link_type type
    int upstream[NUM_UPSTREAM_LINKS];
    int downstream[NUM_DOWNSTREAM_LINKS];
    ...
```