Introduction to ParFlow

What is ParFlow and why would I use it?

ParFlow Short Course

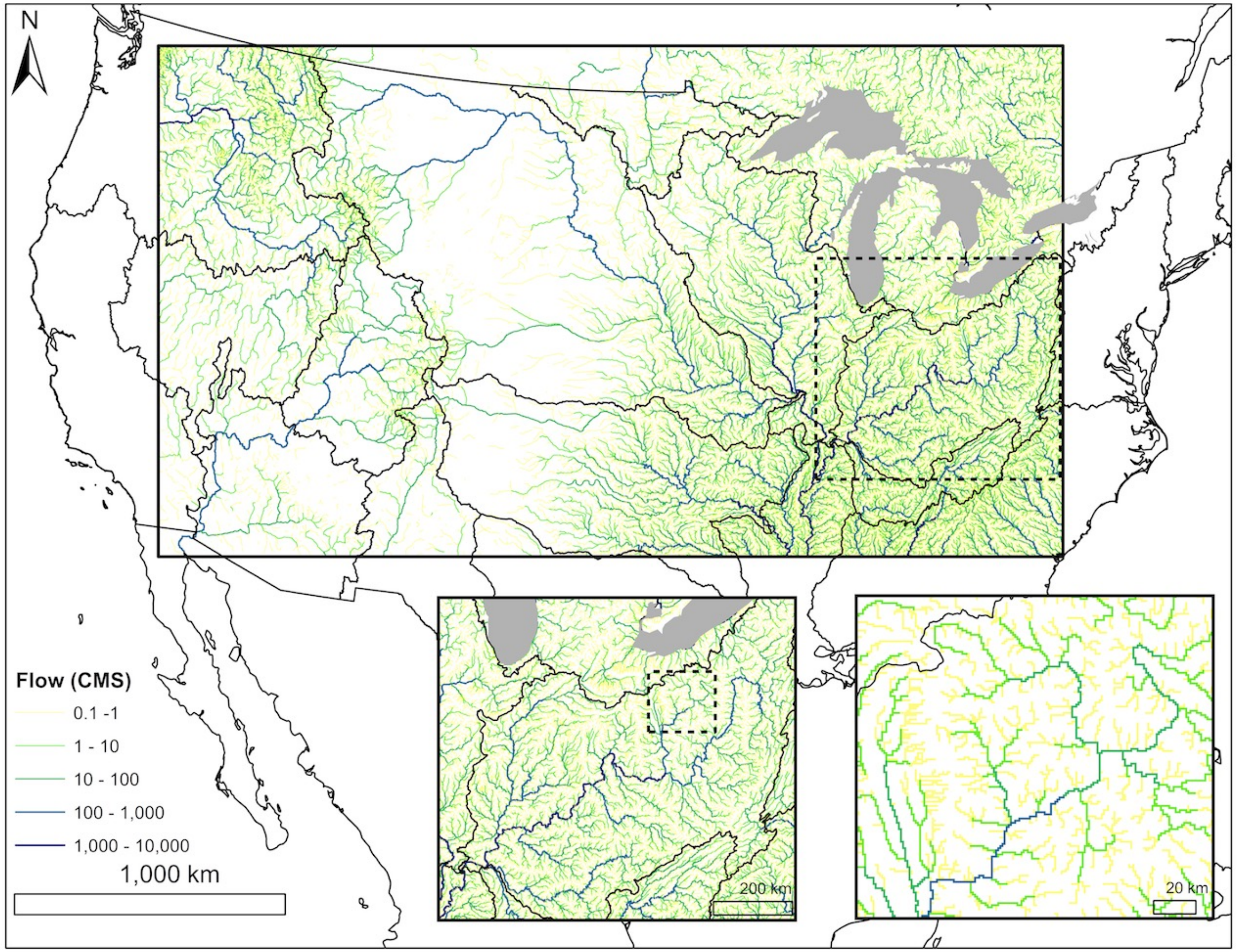
Module 1

Learning Objectives:

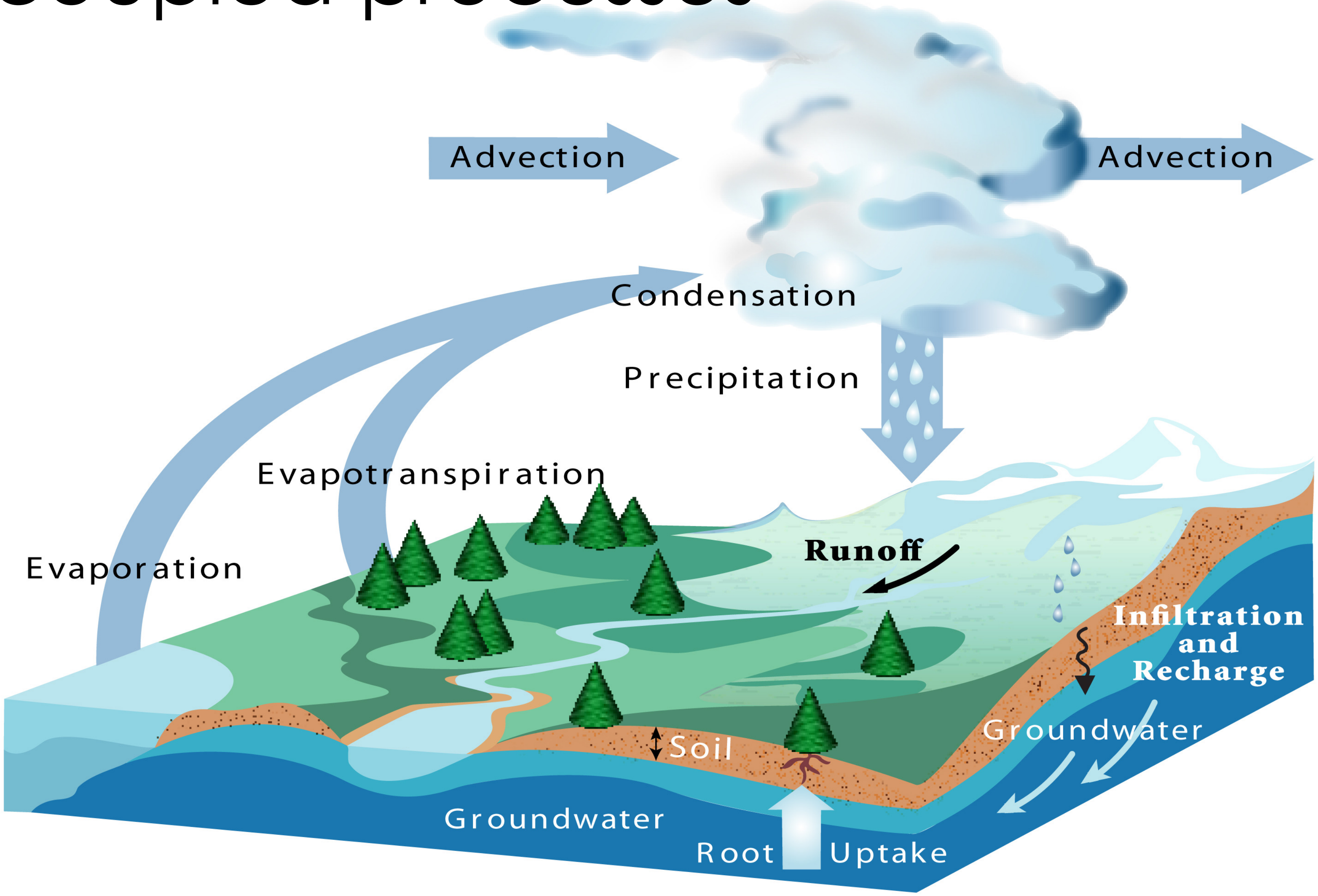
At the end of this module students will understand:

- What is ParFlow and why do we call it an integrated hydrologic model
- What makes the integrated approach different form other groundwater models
- The basics of ParFlow's PDE implementation and parallel design
- The types of problems ParFlow is designed to solve and how it has been used in the past
- Other models that are commonly coupled with ParFlow
- How to contribute to ParFlow's open source development community

The ParFlow Hydrologic Model: A *brief overview*



Terrestrial hydrologic cycle: many coupled processes



The concept for integrated hydrologic models was envisioned half-century ago

Journal of Hydrology **9** (1969) 237-258; © *North-Holland Publishing Co., Amsterdam*

BLUEPRINT FOR A PHYSICALLY-BASED, DIGITALLY-SIMULATED HYDROLOGIC RESPONSE MODEL

R. ALLAN FREEZE

Inland Waters Branch, Department of Energy, Mines and Resources, Calgary, Alberta, Canada

and

R. L. HARLAN

Forestry Branch, Department of Fisheries and Forestry, Calgary, Alberta, Canada

Abstract: In recent years hydrologists have subjected the various subsystems of the hydrologic cycle to intensive study, designed to discover the mechanisms of flow and to

Traditional watershed models are *situation based* and involve multiple *decision points*

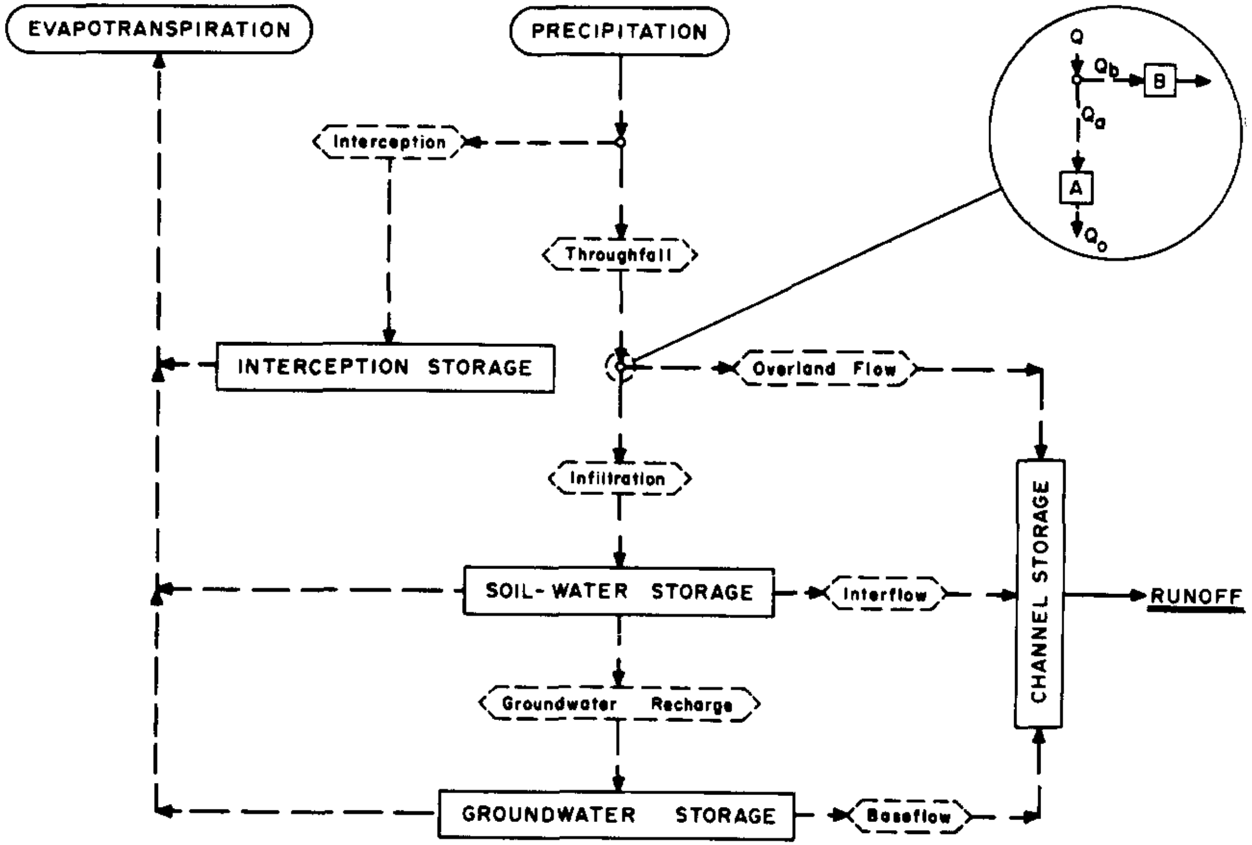
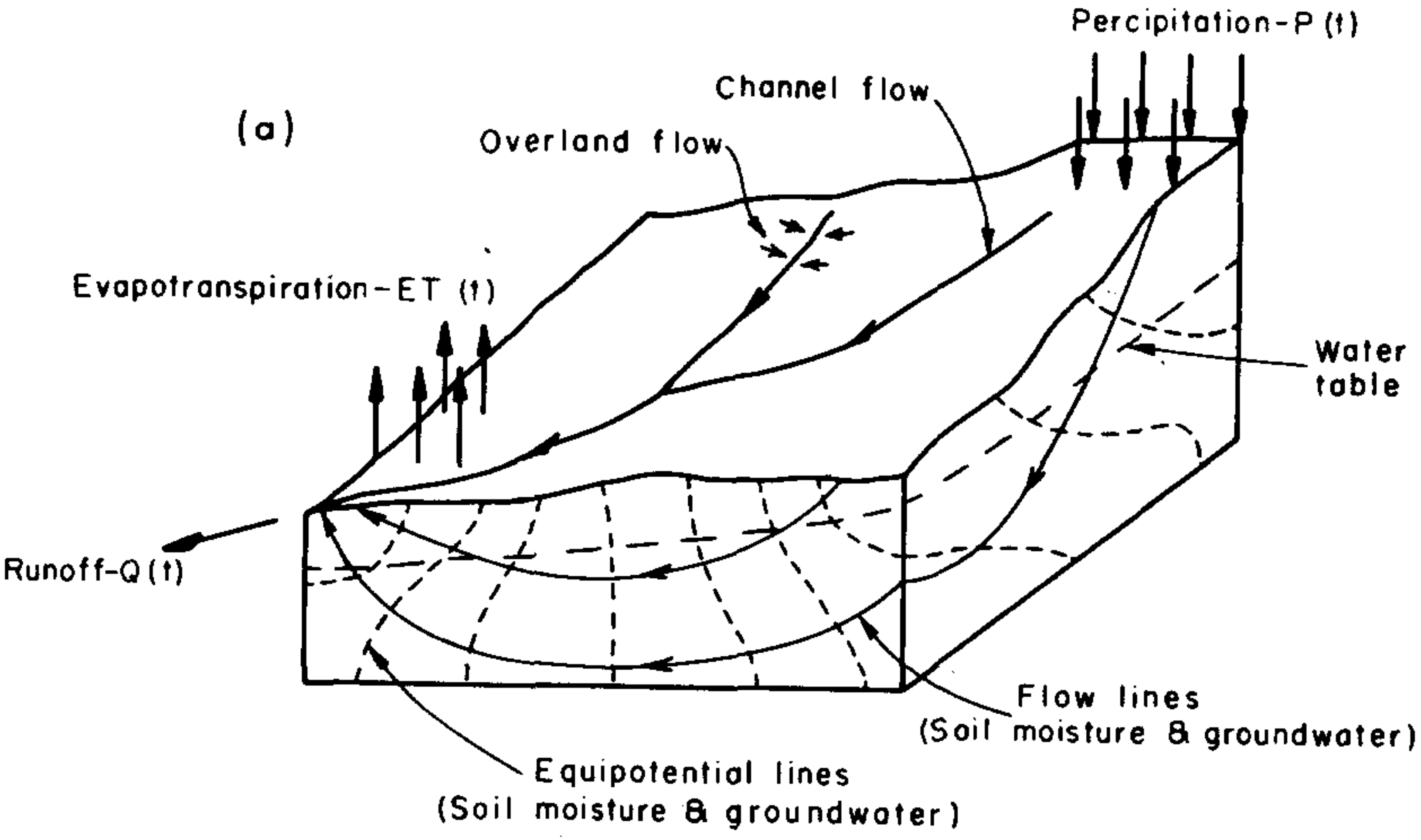


Fig. 2. A conceptual hydrologic model of the type used in the stepwise routing approach of systems hydrology.

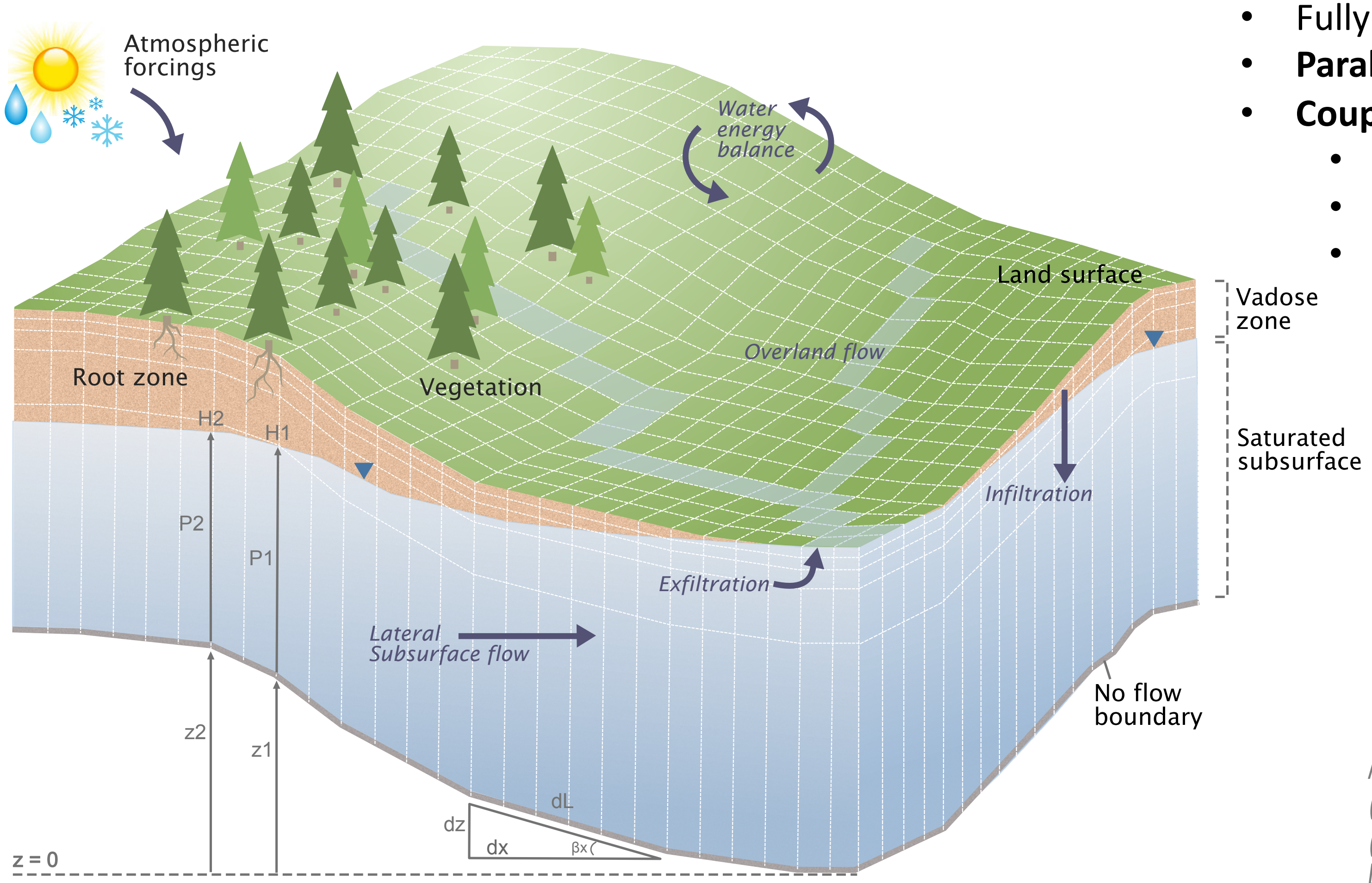
Freeze and Harlan (1969)

They introduced a conceptual model that better represented the way we still envision hydrologic systems



Freeze and Harlan (1969)

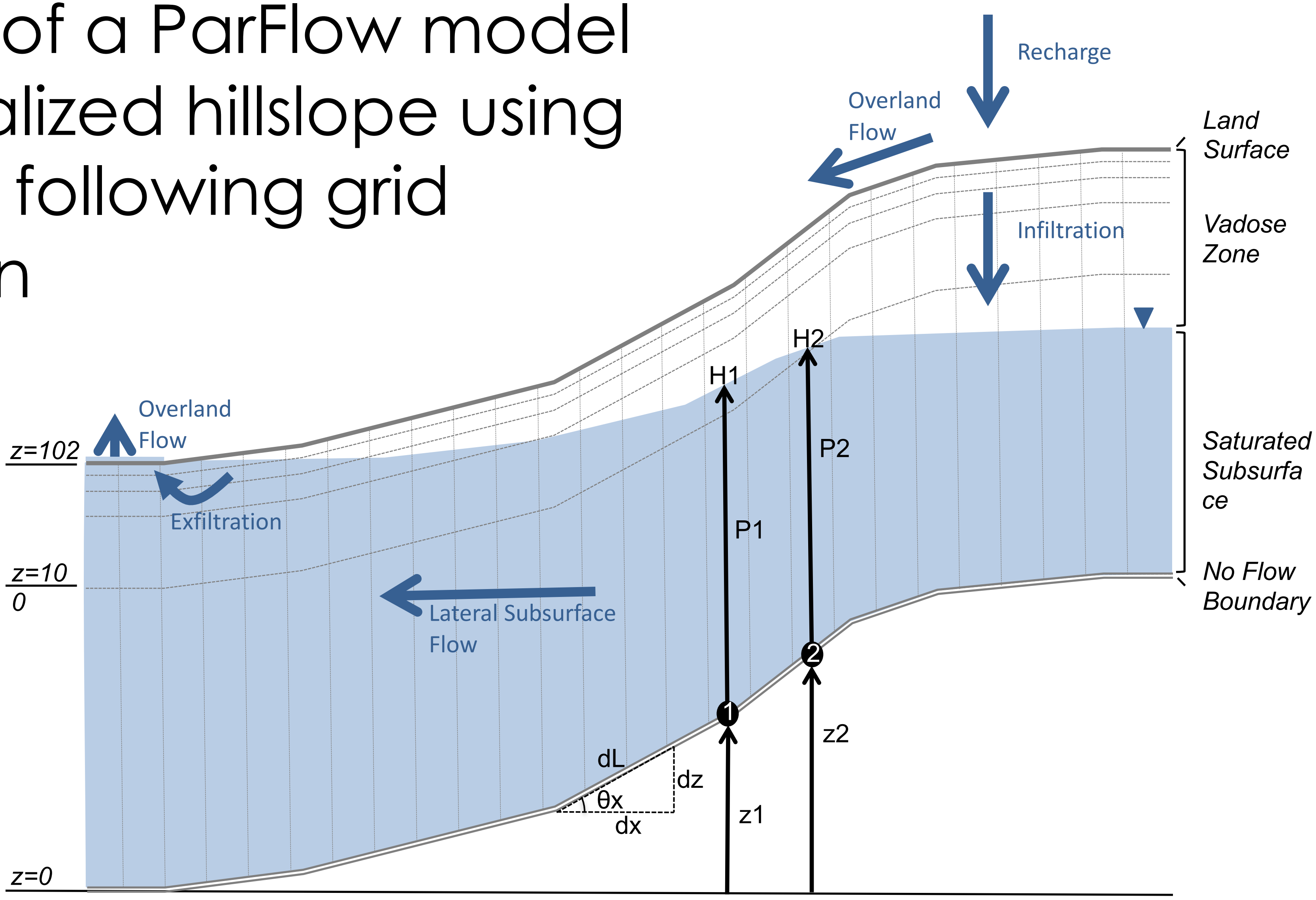
The integrated hydrologic model ParFlow is a tool for computational hydrology



- Variably saturated groundwater flow
- Fully integrated surface water
- **Parallel implementation**
- **Coupled land surface processes**
  - Land-energy balance
  - Snow dynamics
  - Driven by meteorology

Maxwell (2013); Kollet and Maxwell (2008); Kollet and Maxwell (2006);Maxwell and Miller (2005); Dai et al. (2003); Jones and Woodward (2001); Ashby and Falgout (1996)

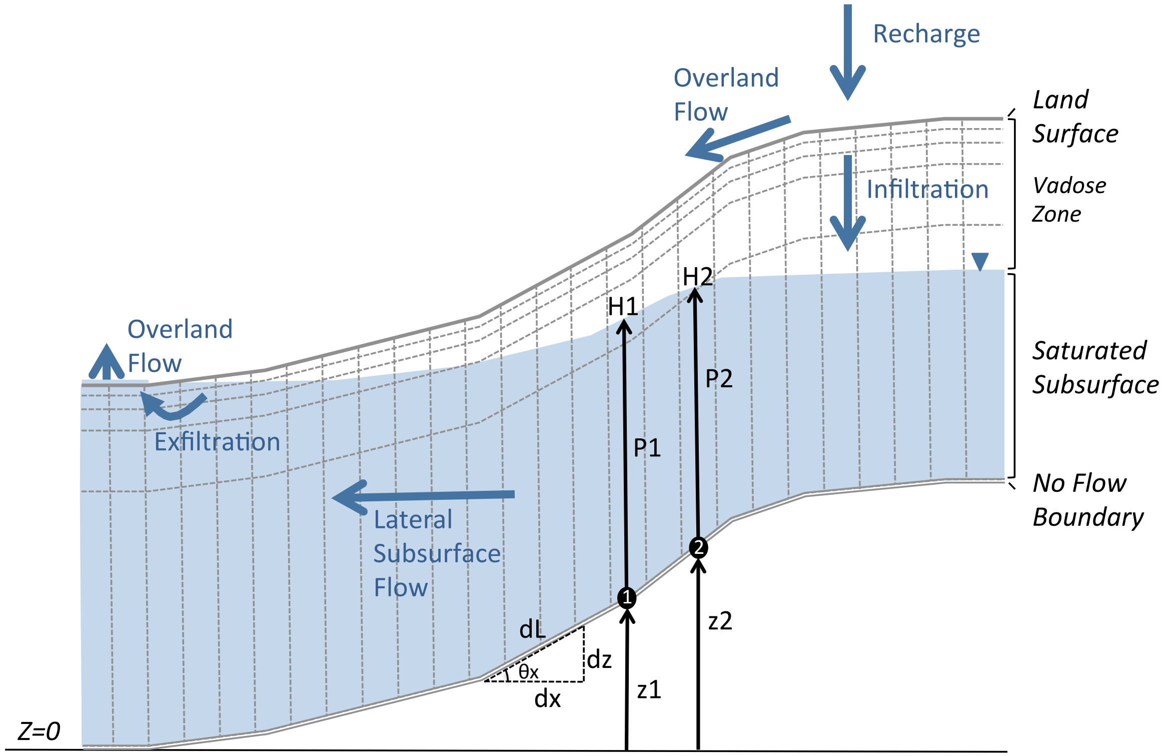
Illustration of a ParFlow model for an idealized hillslope using the terrain following grid formulation



We solve the mixed form of Richards' and Shallow Water equations

$$S_S S_W(h)\frac{\partial h}{\partial t} + \theta \frac{\partial S_W(h)}{\partial t} = \nabla \cdot \mathbf{q} + q_r(\mathbf{x}, z)$$

Mixed form of Richards' we solve for h



We solve the mixed form of Richards' and Shallow Water equations

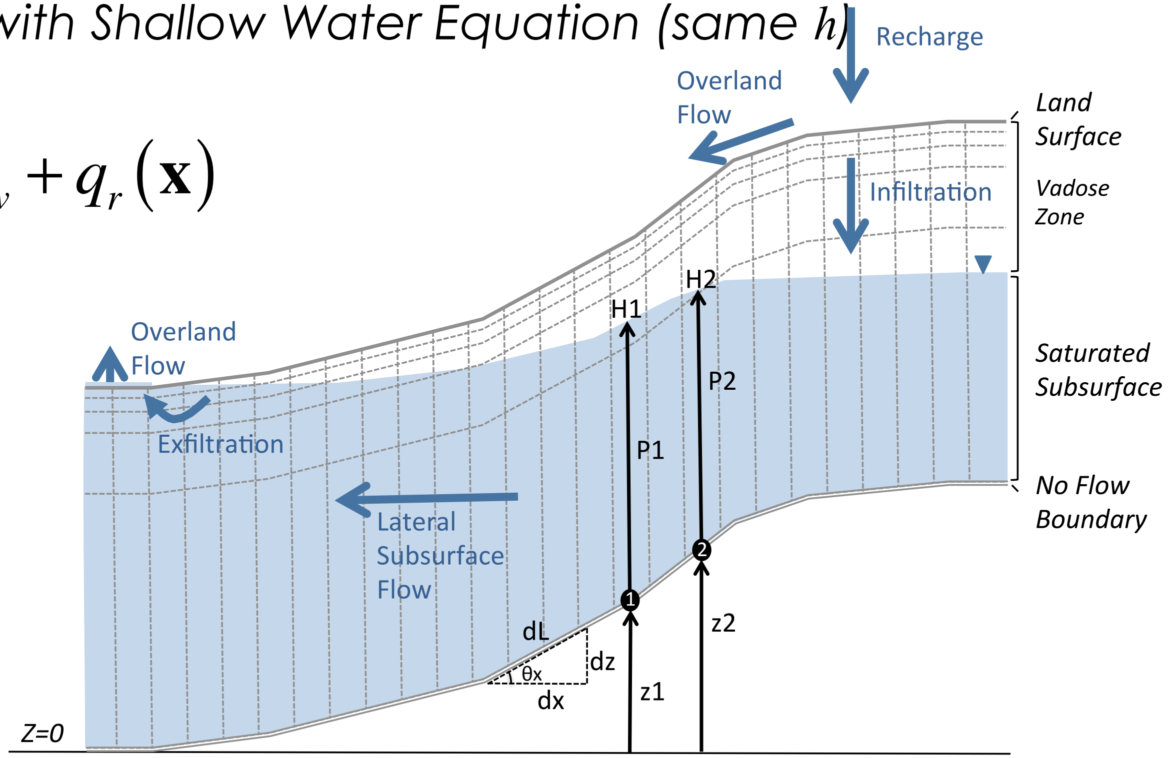
$$S_S S_W(h)\frac{\partial h}{\partial t}+\theta\frac{\partial S_W(h)}{\partial t}=\nabla\cdot\mathbf{q}+q_r(\mathbf{x},z)$$

Mixed form of Richards' we solve for h

Upper boundary condition (Neumann type) combined with Shallow Water Equation (same h)

$$\mathbf{k}\cdot\left(-\mathbf{K}_S(\mathbf{x})k_r(h)\cdot\nabla(h+z)\right)=\frac{\partial\|h,0\|}{\partial t}-\nabla\cdot\|h,0\|v_{sw}+q_r(\mathbf{x})$$

$$v_x=-\frac{\sqrt{S_{f,x}}}{n}\psi_s^{\,2/3}$$



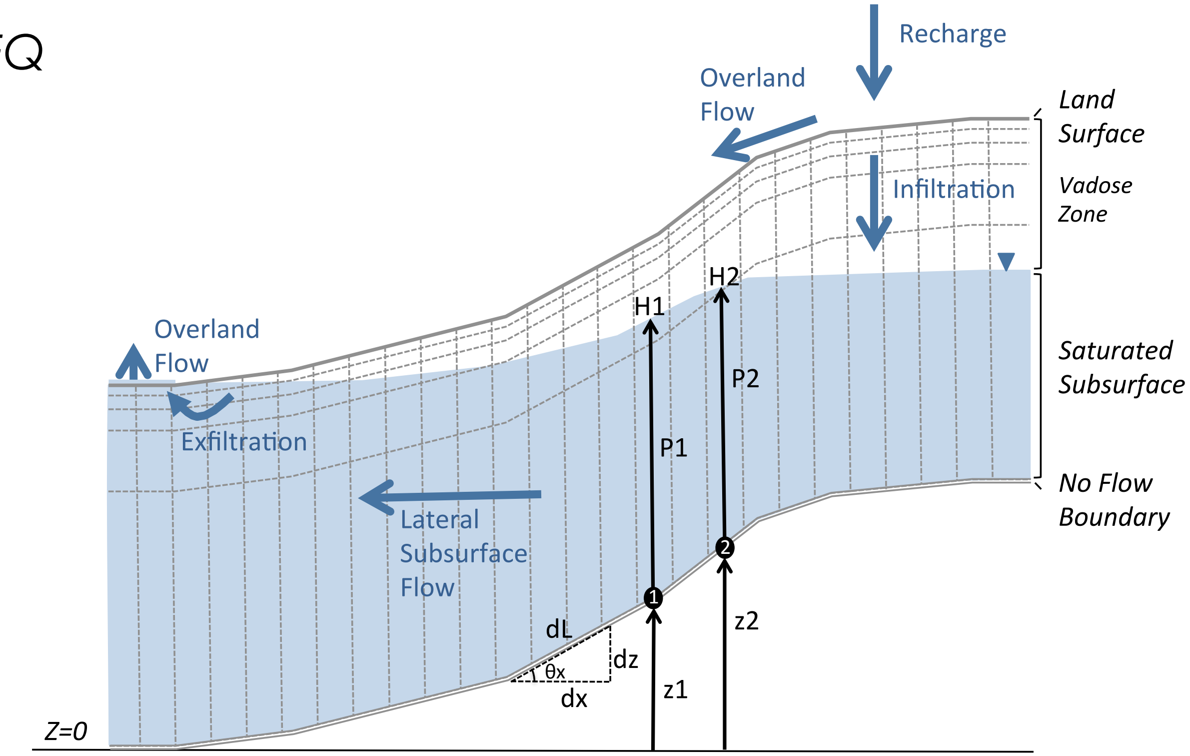
We solve the mixed form of Richards' and Shallow Water equations

$$S_S S_W(h)\frac{\partial h}{\partial t}+\theta\frac{\partial S_W(h)}{\partial t}=\nabla\bullet\mathbf{q}+q_r(\mathbf{x},z)$$

Mixed form of Richards' we solve for h

Flux Relationships from modified Darcy and Mannings EQ

$$\mathbf{q}=-\mathbf{K}_S(\mathbf{x})k_r(h)\left[\nabla(h+z)\cos\beta_x+\sin\beta_x\right]$$



We solve the mixed form of Richards' and Shallow Water equations

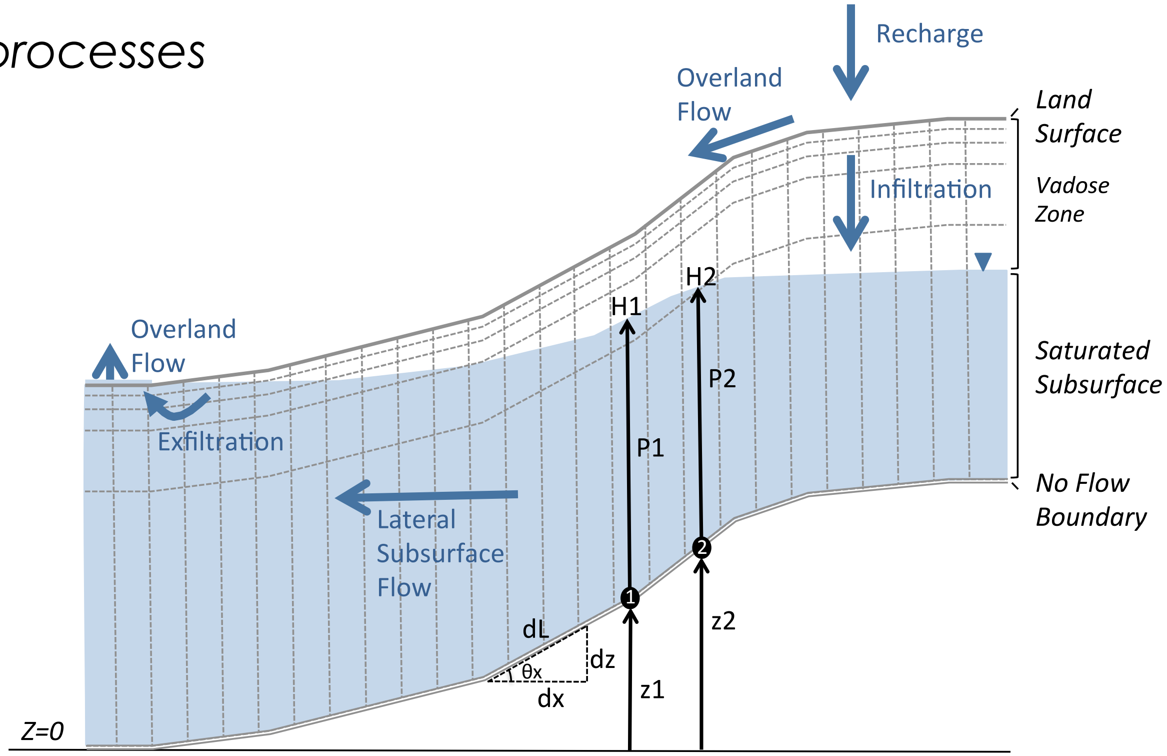
$$S_S S_W(h)\frac{\partial h}{\partial t} + \theta \frac{\partial S_W(h)}{\partial t} = \nabla \cdot \mathbf{q} + q_r(\mathbf{x}, z)$$

Mixed form of Richards' we solve for h

Boundary source/sink, from weather and land surface processes

$$q_r(x,z) = -E_T(x,z)$$

$$q_r(x) = P(x) - E(x)$$



We solve the mixed form of Richards' and Shallow Water equations

$$S_S S_W(h)\frac{\partial h}{\partial t}+\theta\frac{\partial S_W(h)}{\partial t}=\nabla\bullet\mathbf{q}+q_r(\mathbf{x},z)$$

Mixed form of Richards' we solve for h

Upper boundary condition (Neumann type) combined with Shallow Water Equation (same h)

$$\mathbf{k}\bullet\left(-\mathbf{K}_S(\mathbf{x})k_r(h)\bullet\nabla(h+z)\right)=\frac{\partial\|h,0\|}{\partial t}-\nabla\bullet\|h,0\|v_{sw}+q_r(\mathbf{x})$$

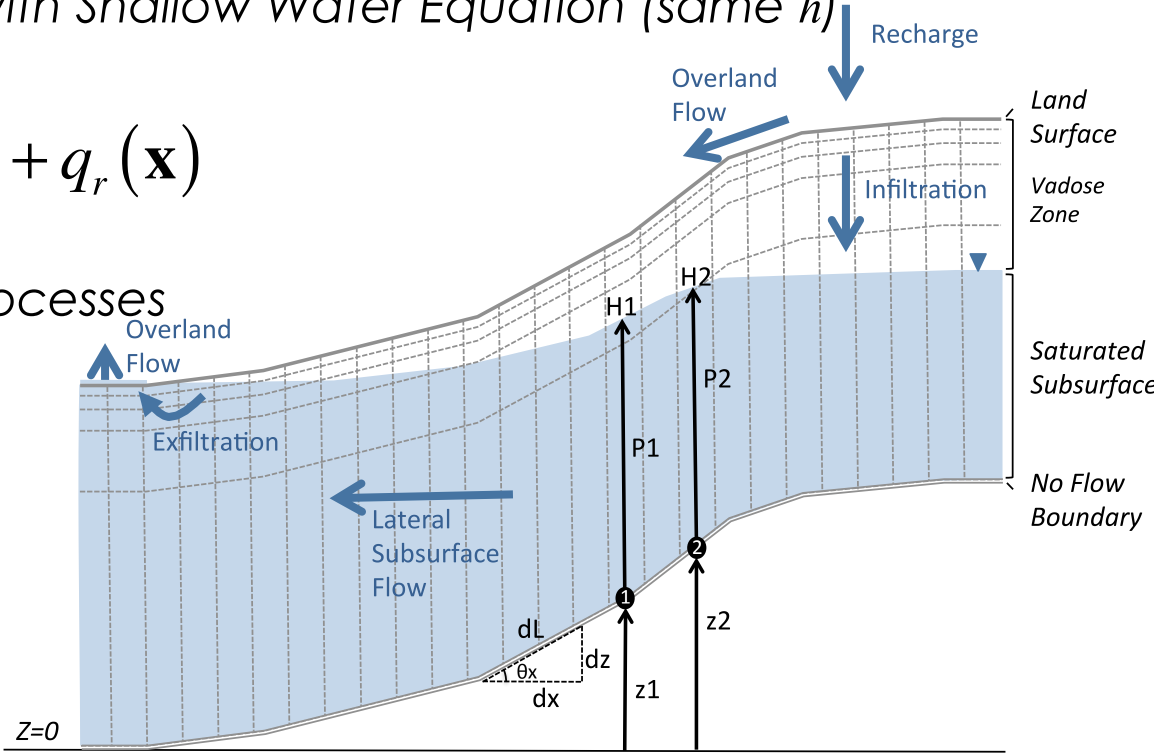
Boundary source/sink, from weather and land surface processes

$$q_r(x)=P(x)-E(x) \qquad q_r(x,z)=-E_T(x,z)$$

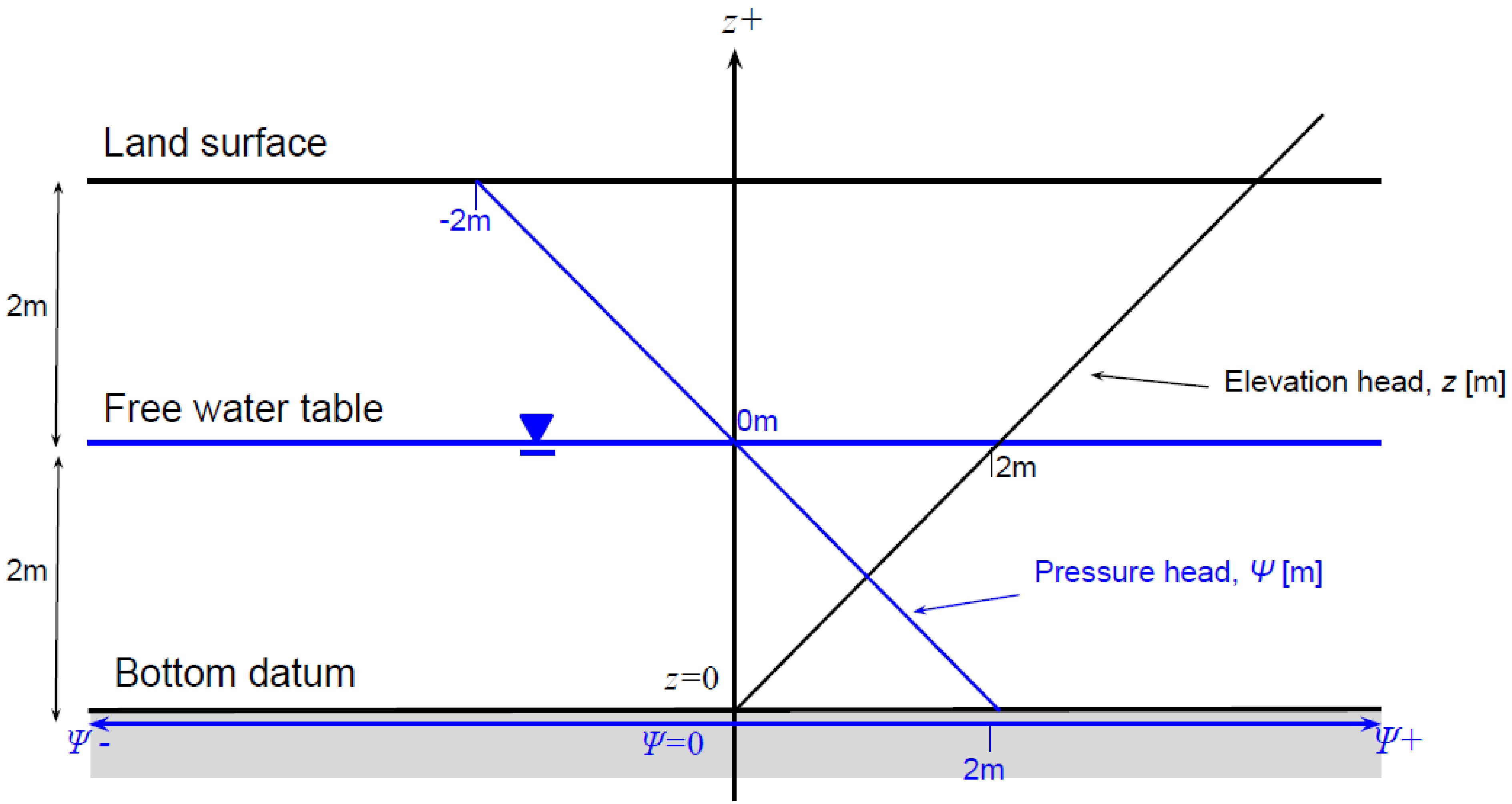
Flux Relationships from modified Darcy and Mannings EQ

$$\mathbf{q}=-\mathbf{K}_S(\mathbf{x})k_r(h)\left[\nabla(h+z)\cos\beta_x+\sin\beta_x\right]$$

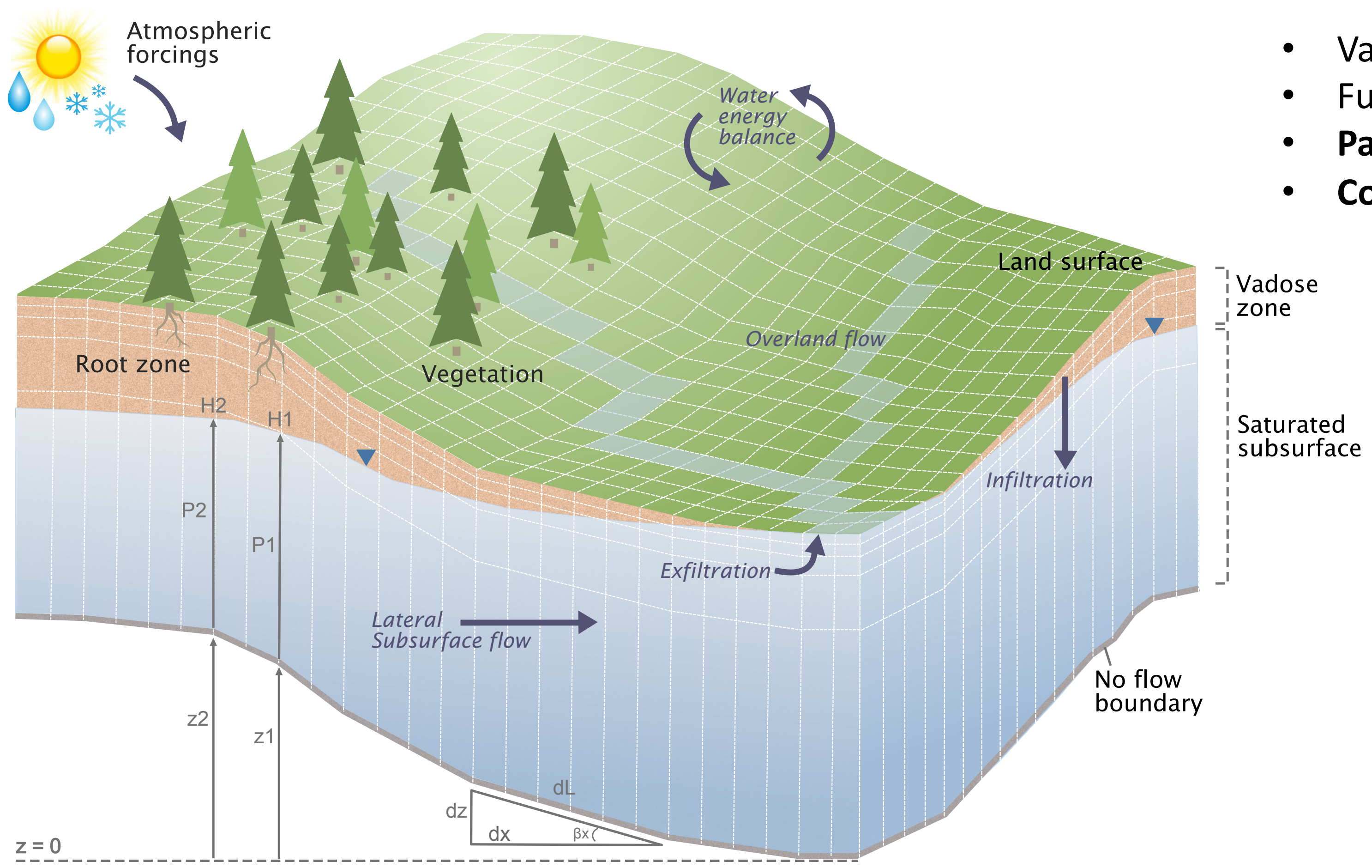
$$v_x=-\frac{\sqrt{S_{f,x}}}{n}\psi_s^{2/3}$$



The hydrostatic profile



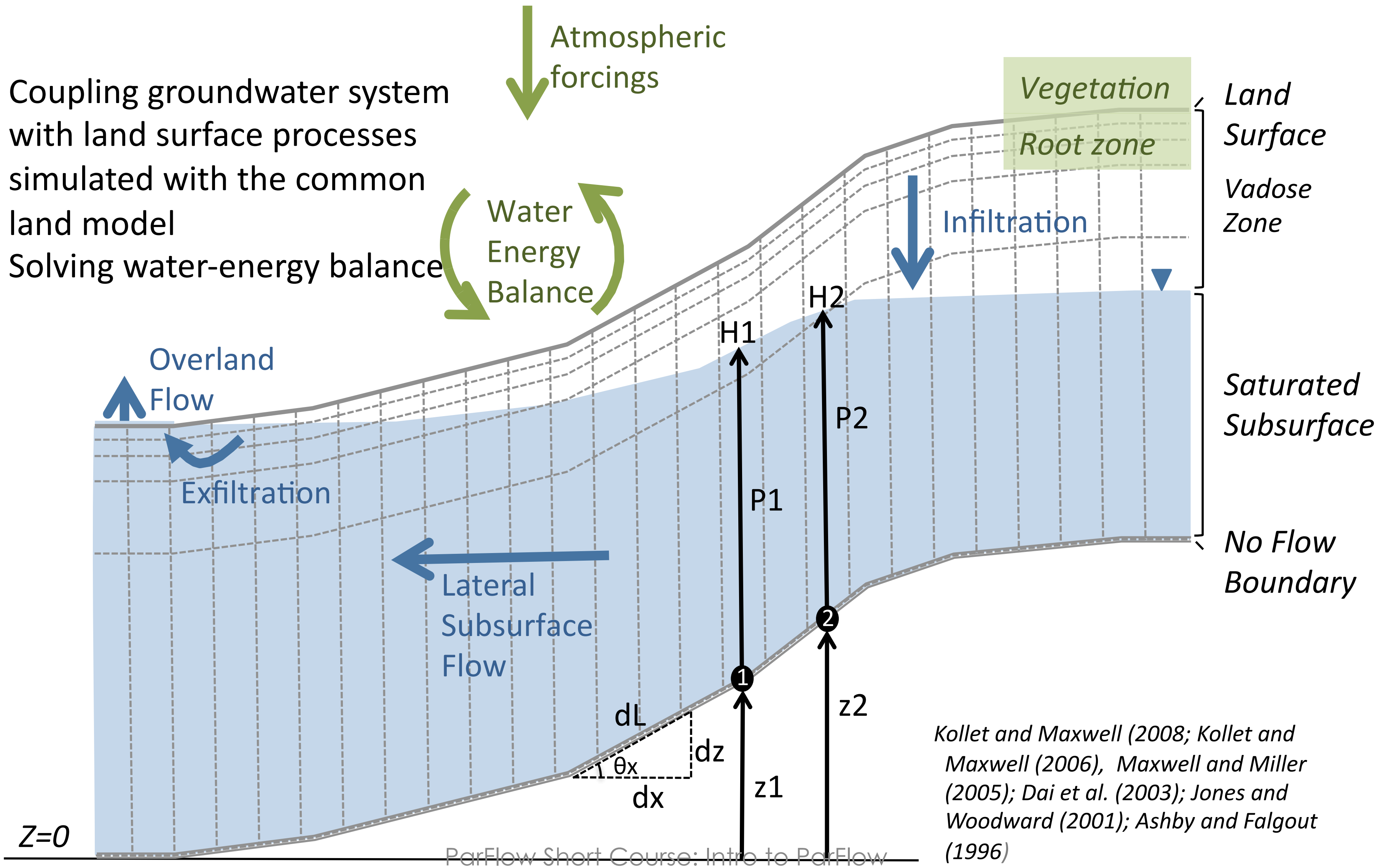
ParFlow-CLM: coupling with land surface processes (CLM) allows for simulation of interactions and connections



- Variably saturated groundwater flow
- Fully integrated surface water
- **Parallel implementation**
- **Coupled land surface processes**
  - Land-energy balance
  - Snow dynamics
  - Driven by meteorology

ParFlow - CLM

- Coupling groundwater system with land surface processes simulated with the common land model
- Solving water-energy balance



Kollet and Maxwell (2008; Kollet and Maxwell (2006), Maxwell and Miller (2005); Dai et al. (2003); Jones and Woodward (2001); Ashby and Falgout (1996)

The model input structure is a novel aspect

- TCL/TK or Python scripting language
- All parameters input as keys
- Keys used to build a database that ParFlow uses
- ParFlow executed by as MPI
- Since input file is a script may be run like a program, in a notebook, etc

Python input script:
Set database keys for simulation, any other manipulations.

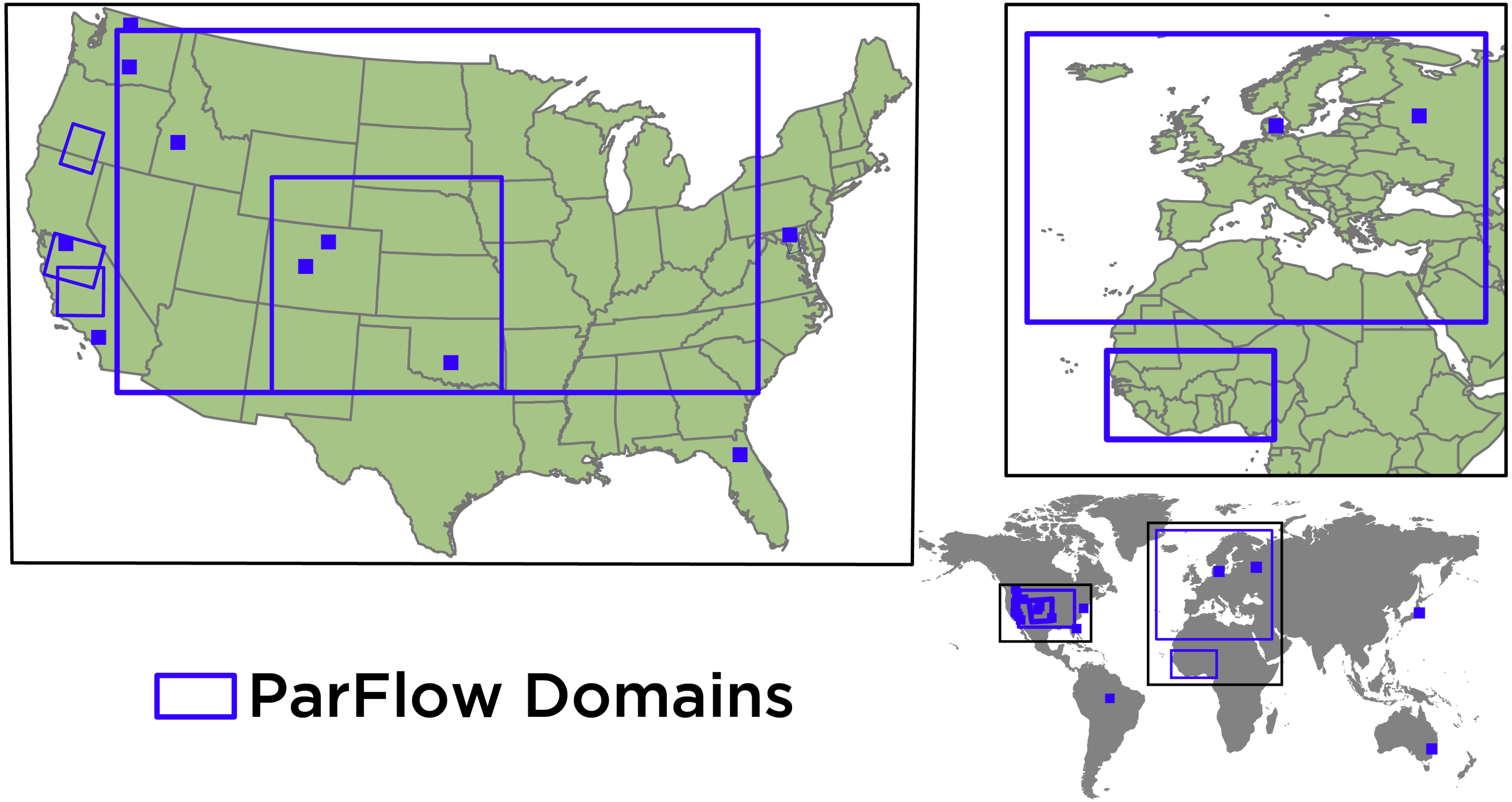
→

`Model.run()` command:
1. Executes `parflow.ipynb` script
2. Write database (`.pfidb`) file
3. Set up parallel run parameters
4. Execute `run` script

→

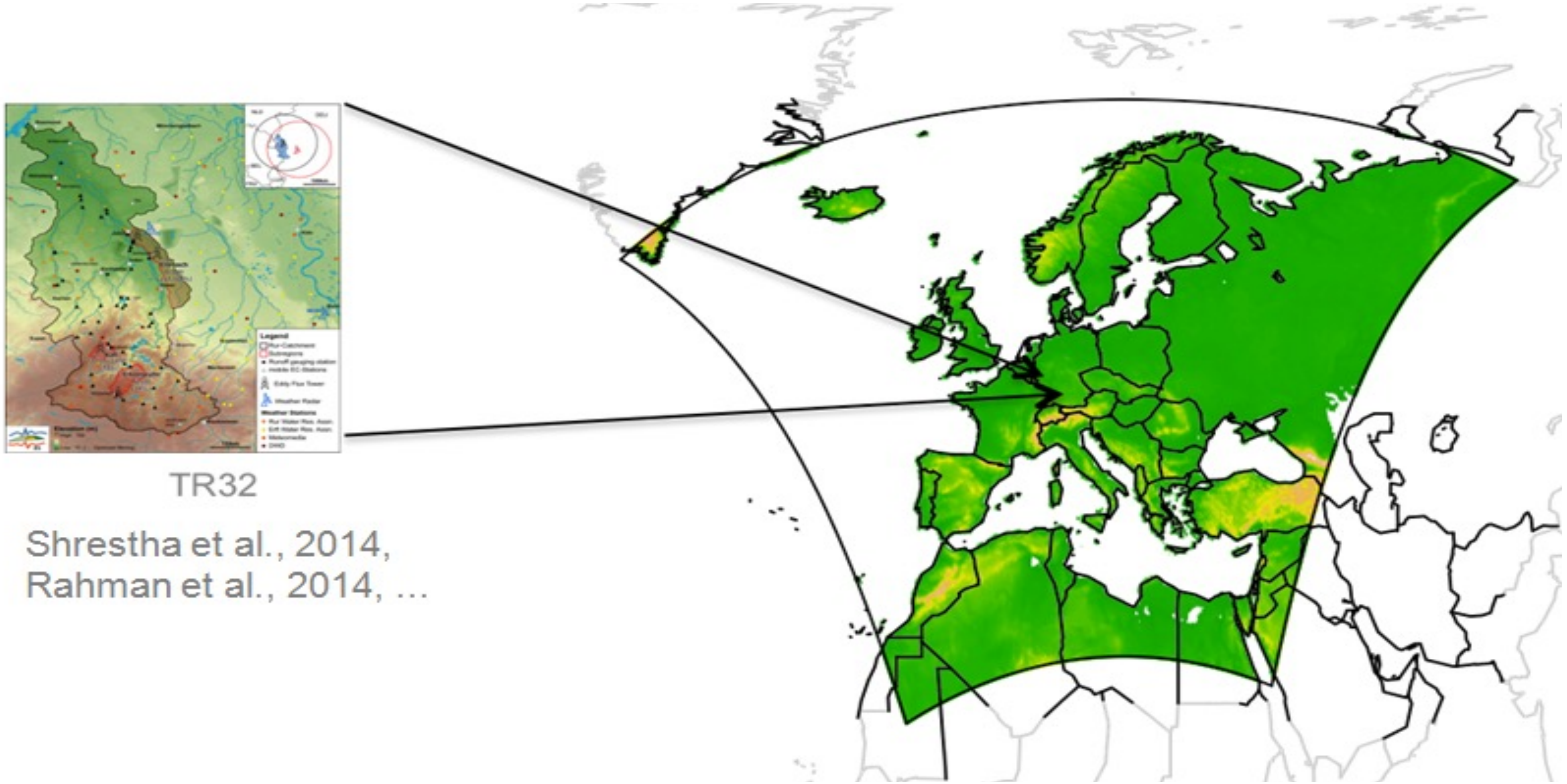
`run` script:
1. Execute ParFlow using platform specific options
2. Port standard output to a file

ParFlow has been applied at many sites worldwide



ParFlow Domains

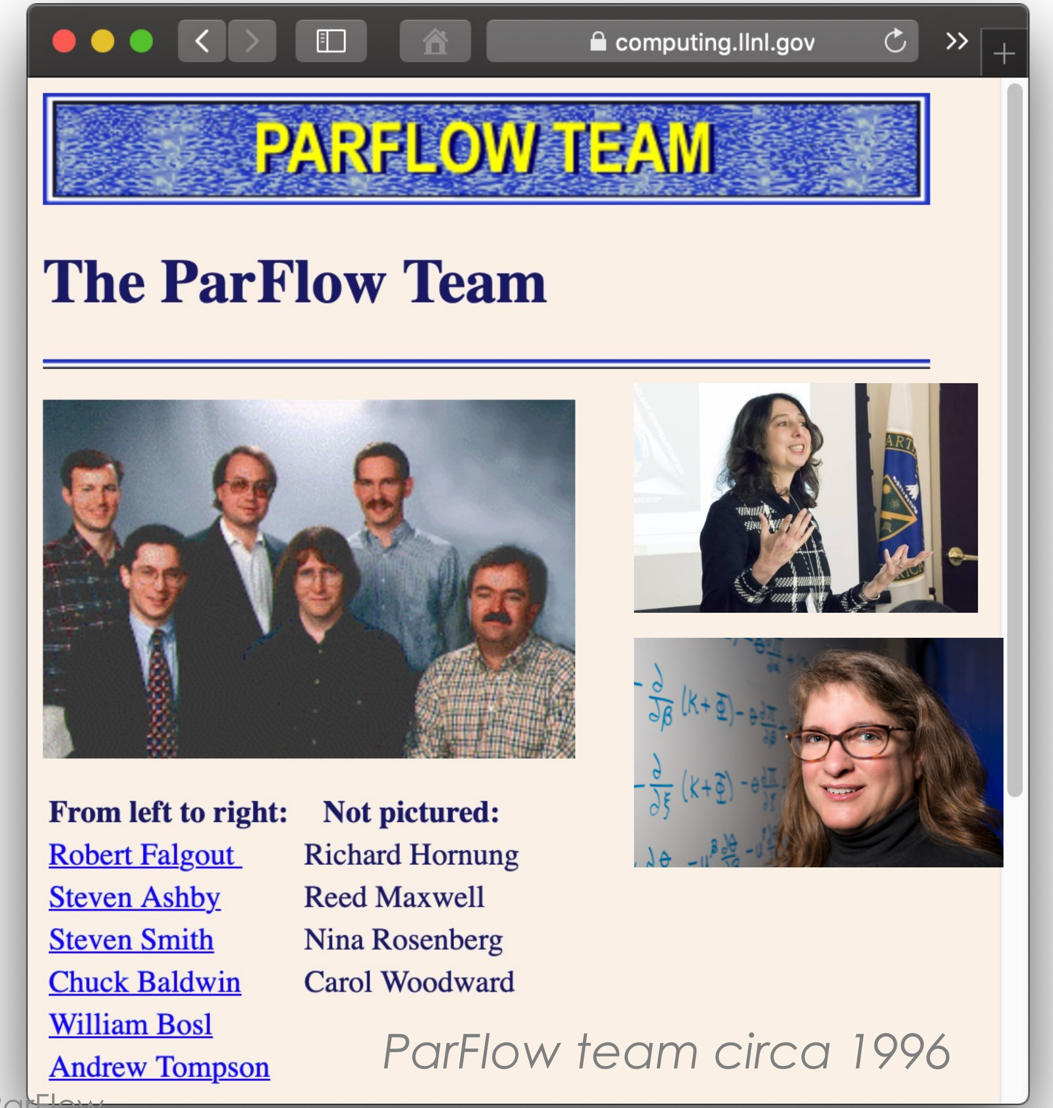
ParFlow has been used to model watersheds around the world at a wide range of scales



Including Big Thompson (CO), Chesapeake (MD), CONUS, CORDEX, East Inlet (CO), East River (CO), Klamath (OR), Little Washita (OK), Rur (Germany), San Joaquin (CA), Sante Fe (FL), Skjern (Denmark), Wüstebach (Germany)

ParFlow has a long development history

- *Ashby and Falgout 1996*, parallel multigrid saturated flow
- *Jones and Woodward 2001*, parallel Richards' equation flow
- *Maxwell and Miller 2005*, CLM coupling
- *Kollet and Maxwell 2006*, parallel overland flow
- *Maxwell 2013*, Terrain following grid

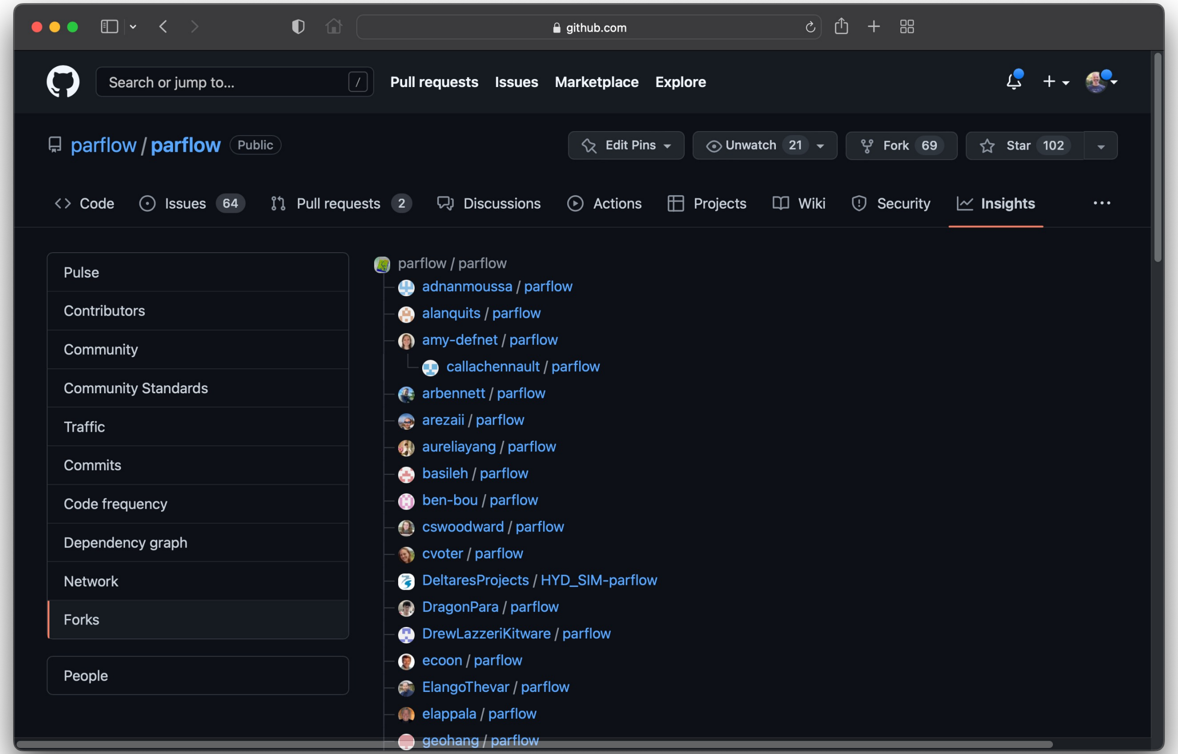


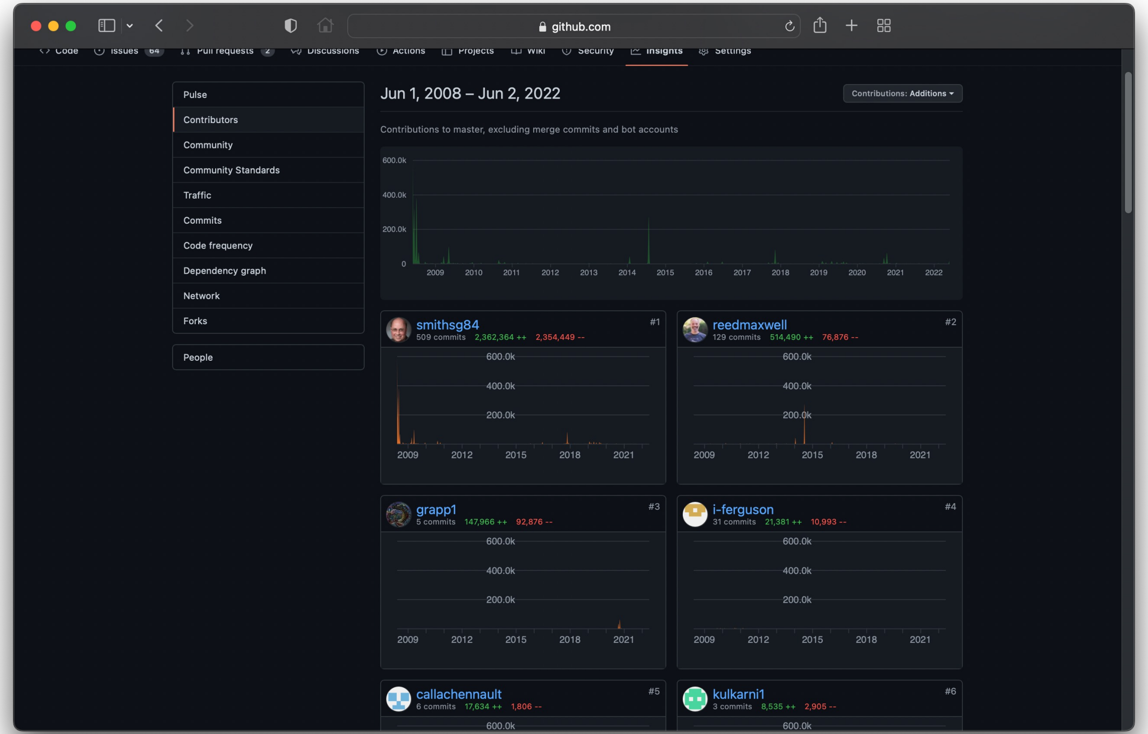
ParFlow team circa 1996

ParFlow is a community code, developed by several groups

- Started at Lawrence Livermore National Laboratory (LLNL)
- Became open source in 2006
- Center of development moved to Colorado School of Mines in 2009
- Now active development from groups in Princeton, University of Arizona, F-Z Jülich, UniBonn, LLNL, LBL, LTHE and WSU, Boise State

ParFlow is a community code, developed by several groups

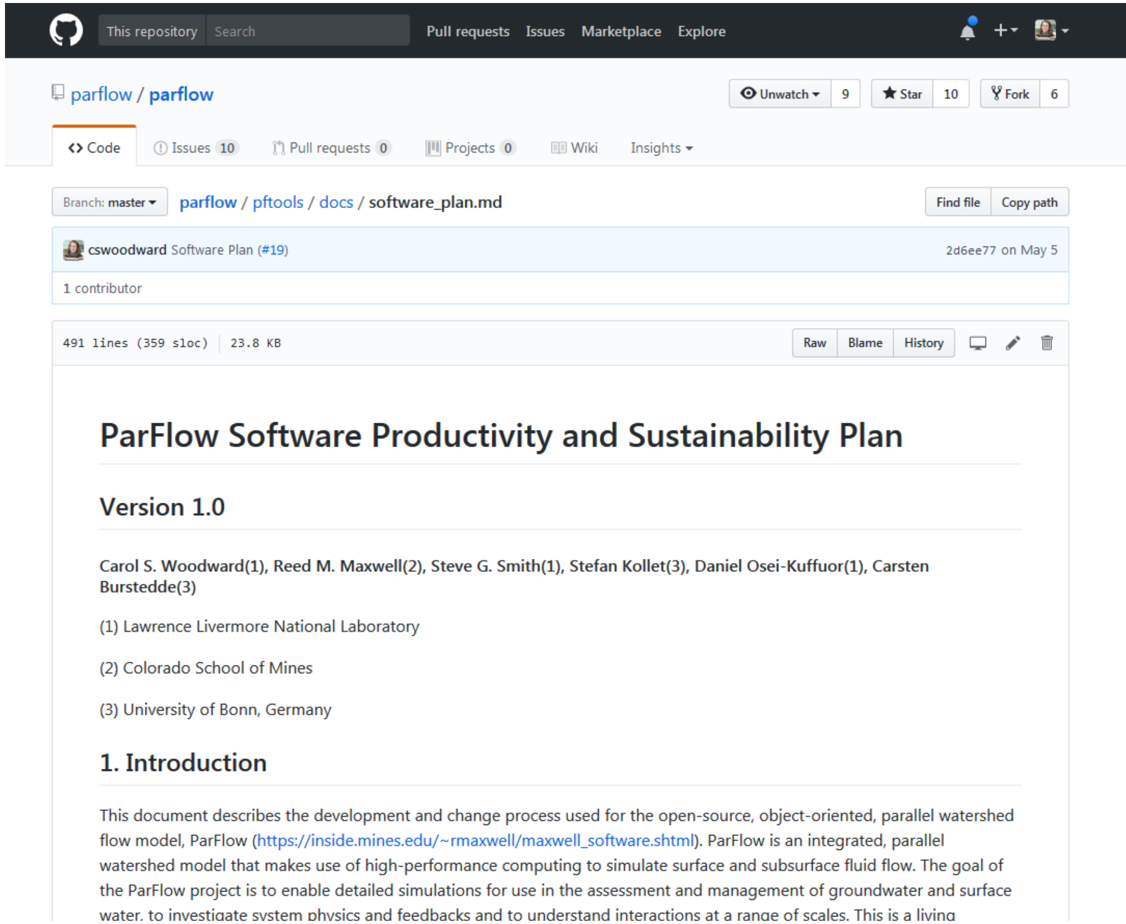




We developed a software productivity and sustainability plan for ParFlow

- This plan lays out formal processes that contributors to ParFlow are expected to follow
    - Requirements for design documents and project review of new feature and capability enhancements
    - Establishes a Git workflow for submission of new code and its review
    - Establishes documentation requirements for new code
    - Describes testing practices
    - Establishes practices for external package usage and testing
- This document helps prevent conflicting features, poorly designed, and untested code from creeping into ParFlow
- This is a living document, expected to be updated and changed as ParFlow continues to evolve

https://github.com/parflow/parflow/blob/master/pftools/docs/software_plan.md



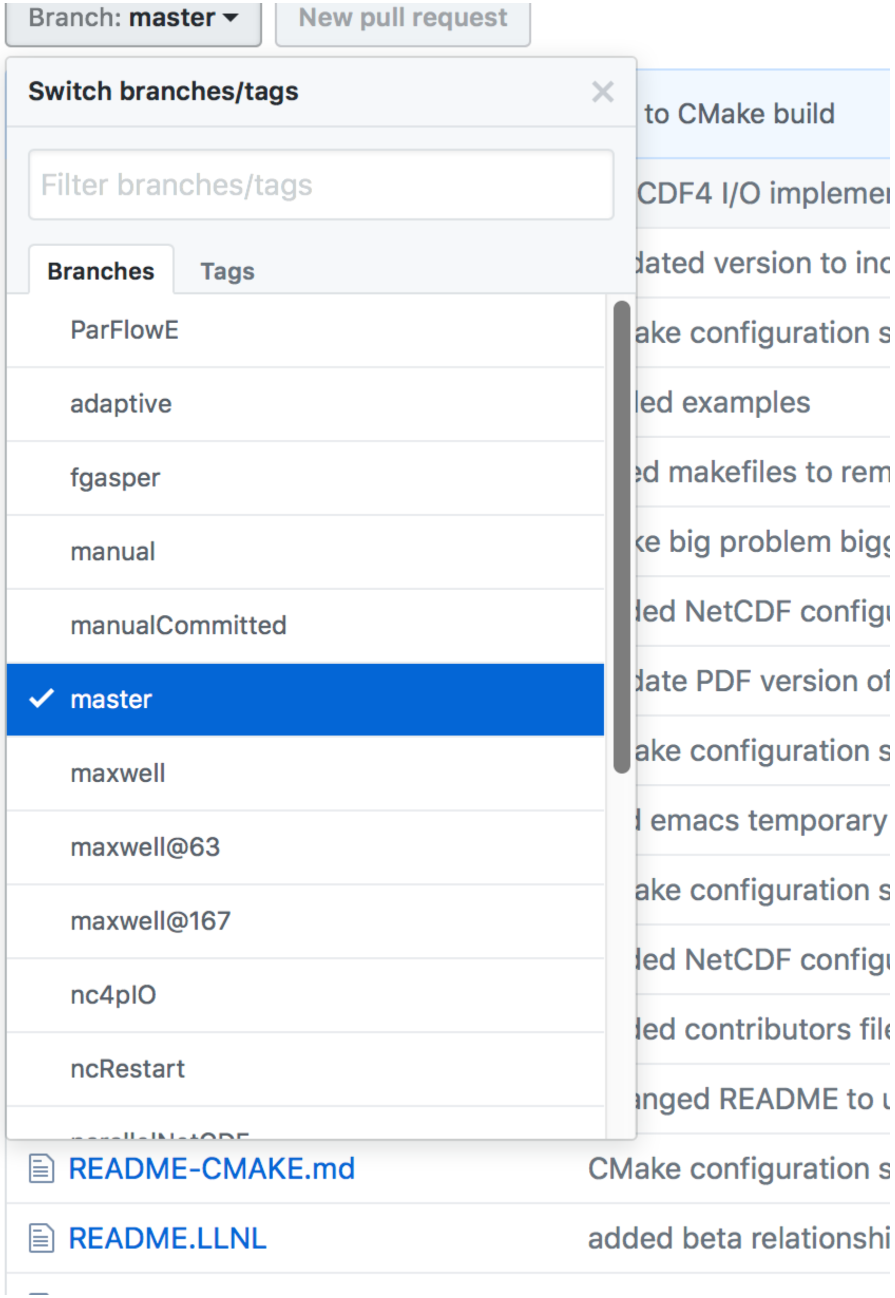
Centralized version control and development is critical

- ParFlow is hosted on GitHub
  - Git repository used for
    - Source code
    - Test cases
    - Documentation
  - Issue tracking
    - Not heavily used yet
    - Transitioning users to submitting issues rather than emails for better tracking

https://github.com/parflow

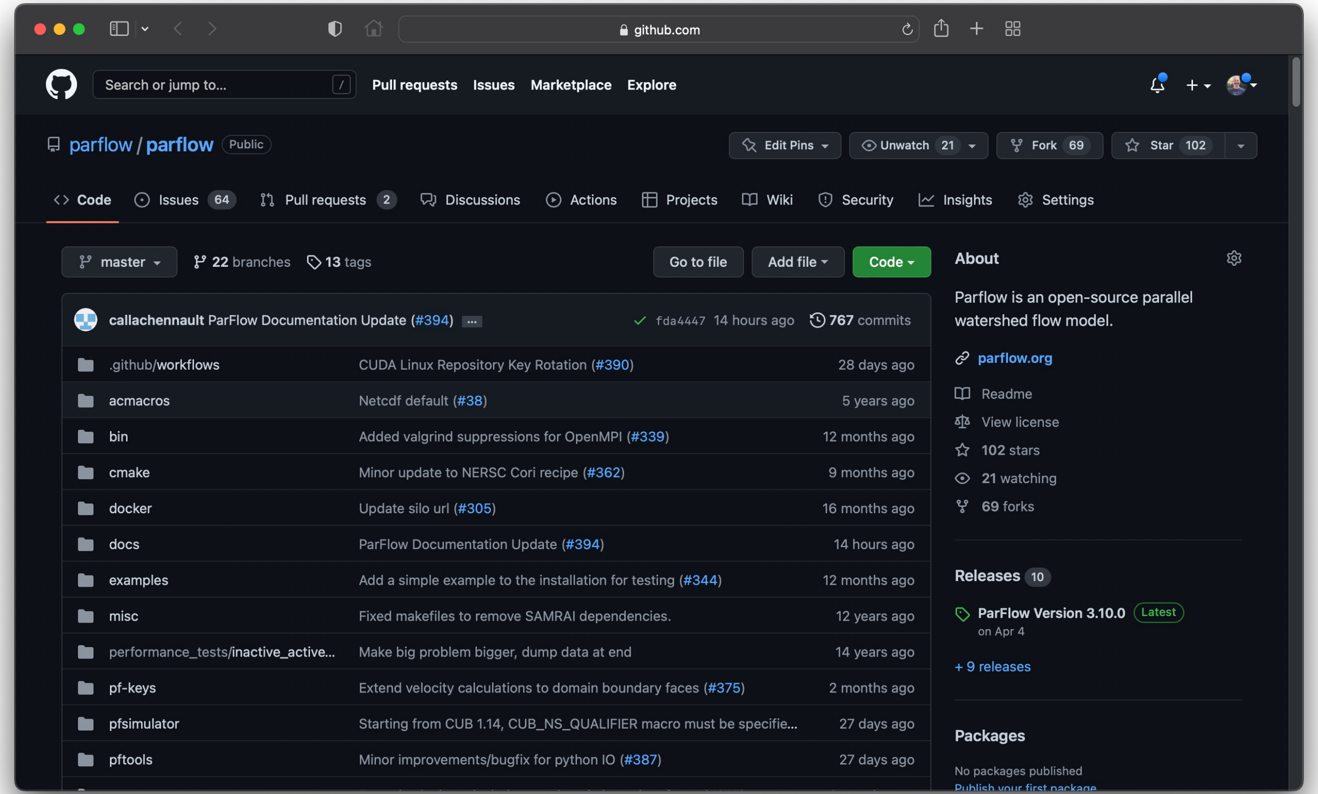
With many active development branches the development plan is important

- **Branches** span range and scope
- New **features** must play well with each other
- **Versioning** and backward compatibility



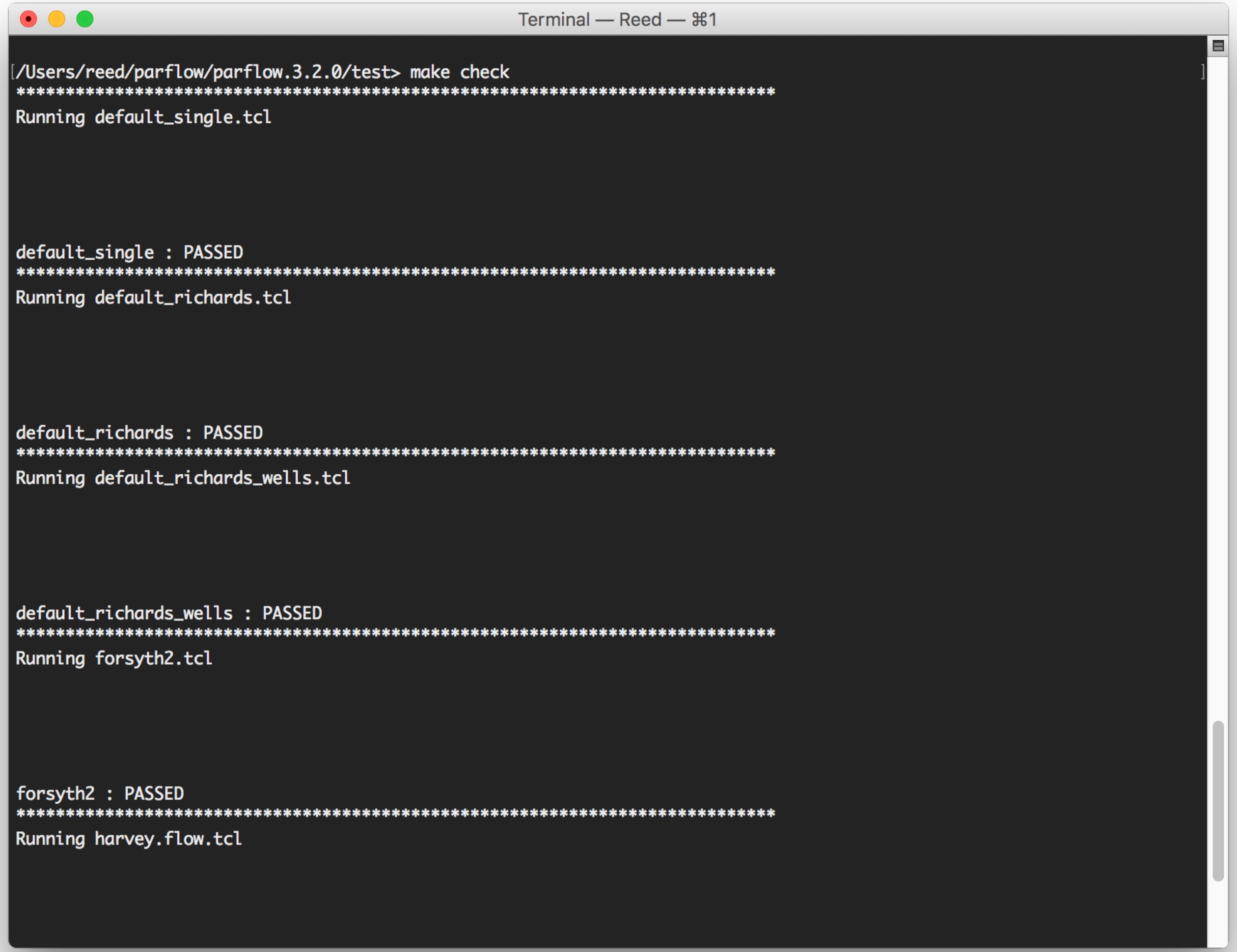
GitHub provides a mechanism to enforce the development plan

- Standard GitHub Pull request model is used for code submissions
  - Development is done on branches or GitHub repository forks
    - Only lead maintainers have write access to master branch
  - Pull requests are reviewed by lead maintainers
    - Correctness
    - Consistency with coding style
    - Verify test for new feature was added
    - Regression tests pass

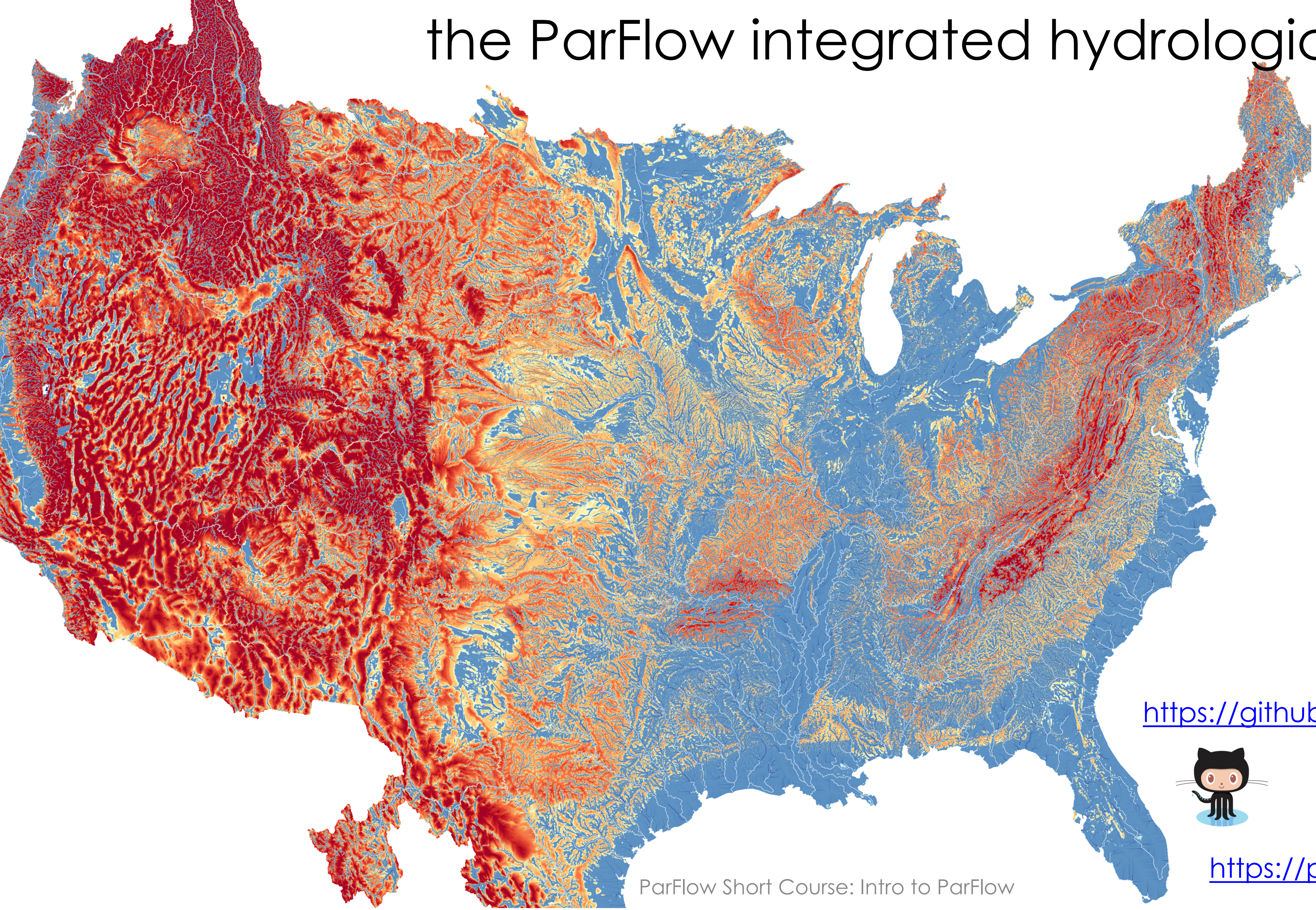


Automated test cases provide code verification mechanism

- Suite of test cases testing solvers and code features
- Contributed with each new major update
- Testing allows check for backward compatibility of new code
- Tests provide examples for new users



the ParFlow integrated hydrologic model



open source
community driven
laptop to supercomputer
point to global scale
Pythonic
portable
CPU and GPU

For More Info

https://parflow.org

https://github.com/parflow/parflow

https://parflow.readthedocs.io