ParFlow Short Course

Princeton University
June 2-3, 2022

This material and short course was supported by the US and German National Science Foundations, the US Department of Energy, and the German Helmholtz Association.

Meet your instructors

Nick Engdahl, Washington State University

Stefan Kollet, FZ-Jülich

University of Bonn

Reed Maxwell,

Princeton University

Laura Condon, University of Arizona

Model is an ambiguous term

- What do we mean when we say *model*?
- Name some models.

There are many types of models

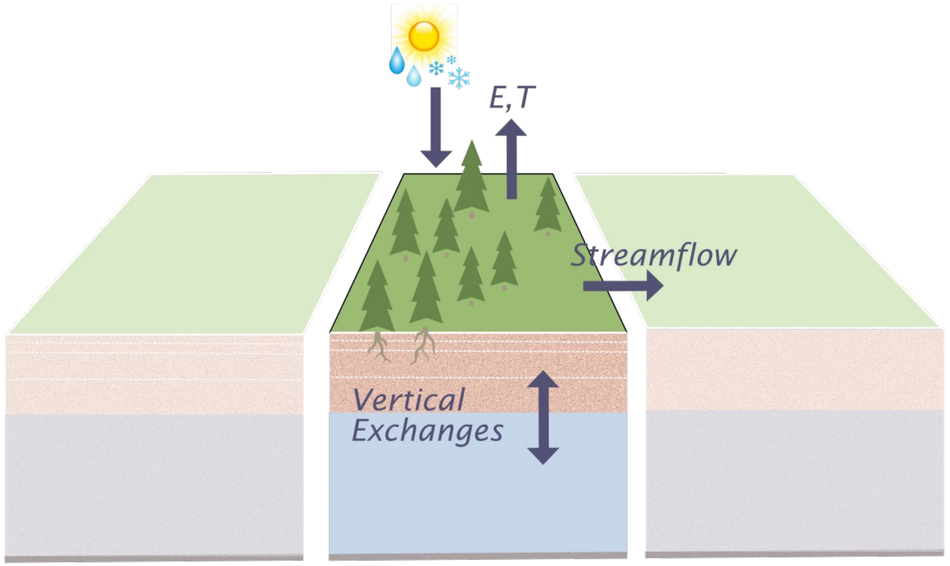
- Conceptual Model
- Physical Model
- Numerical Model
- ML/DL Model

All models have many assumptions

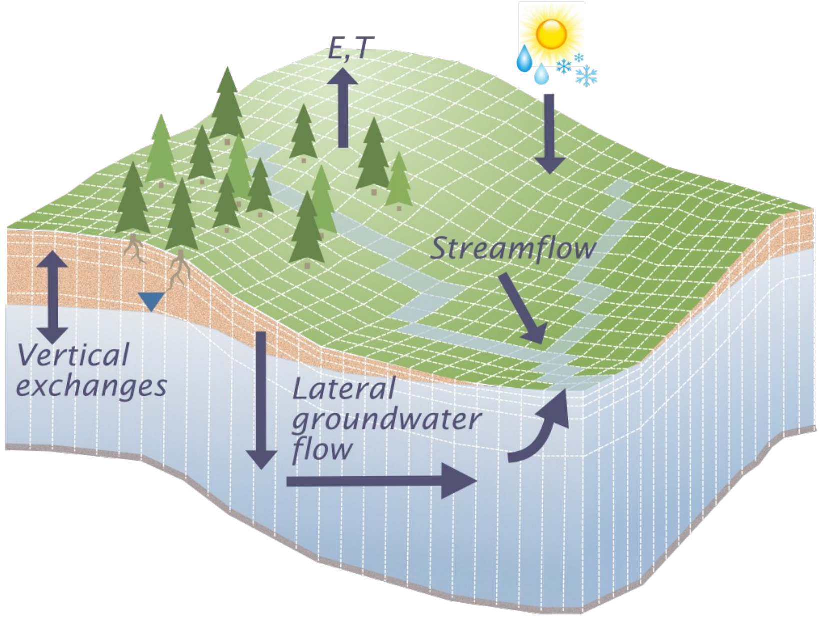
No model is *perfect*

The process of modeling is really important

There are different approaches to modeling complex systems and it is unclear which is ideal



Traditional land surface models **simplify** processes but are **efficient**



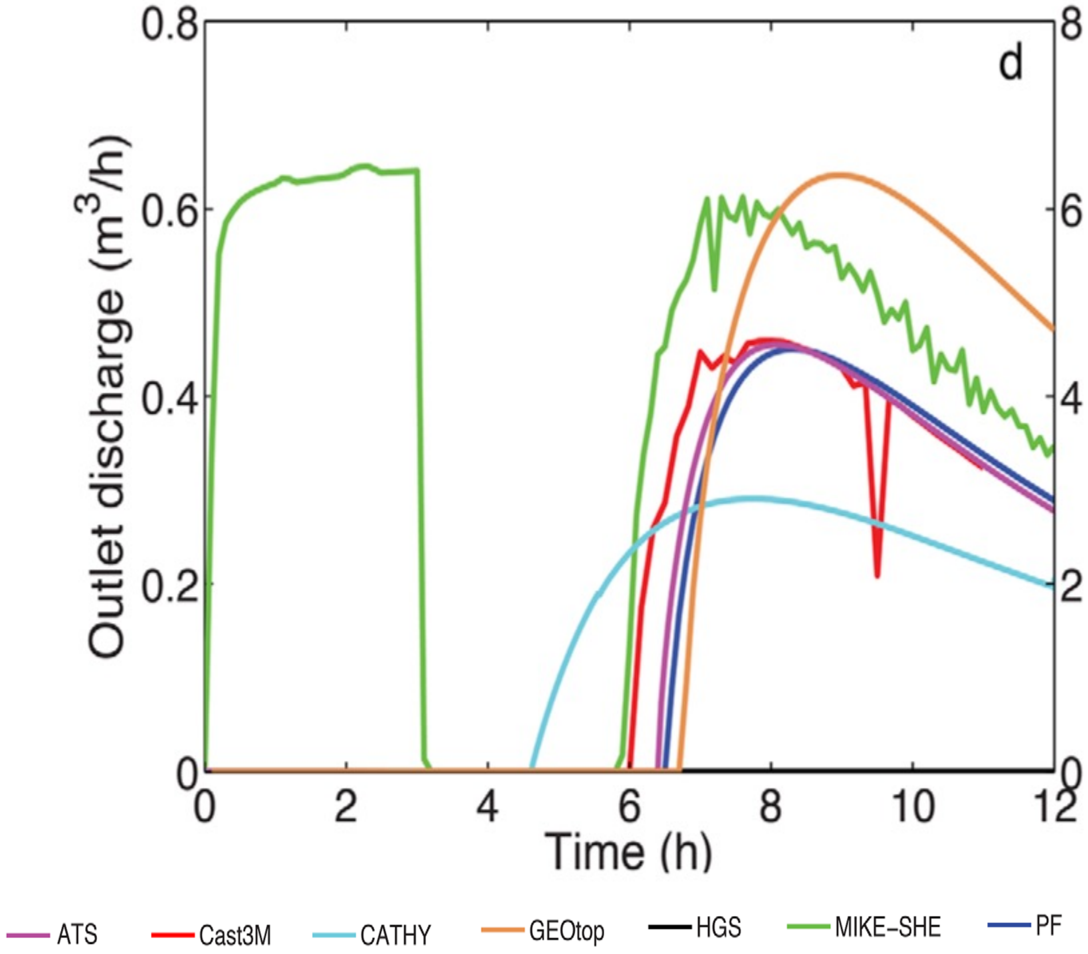
Integrated hydrologic models are **comprehensive** but **expensive**

Computational hydrology is hard but is our best way to relax assumptions

- we solve systems of **nonlinear PDEs**
- we use and develop **robust** and efficient **parallel solution techniques**
- **real problems** with land surface processes are very **hard to solve**

For some problems, this may make the difference between getting the right answers for the right reasons and not

Simulation results from the *superslab* test case show disagreement between conceptual approaches



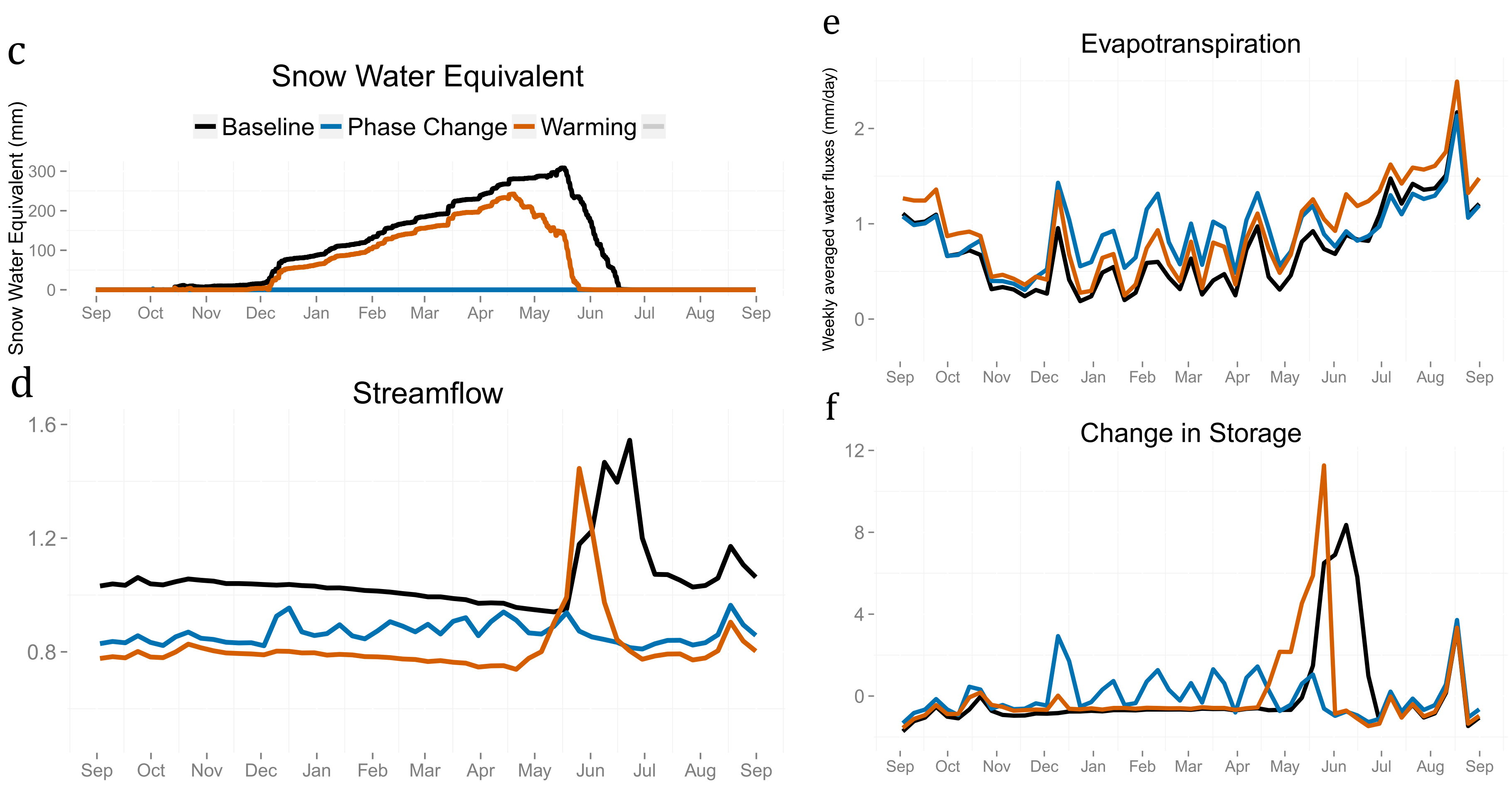
Kollet et al *WRR* 2017

Models can be useful tools to provide insight

- Controlled **numerical experiments** elucidate process interactions under change
- A single perturbation (e.g. temperature increase) can be tracked through the **entire nonlinear system**
- **Connections** we see in simulations can provide **insight** and guide **observations**

Models are useful in ways that the real world isn't

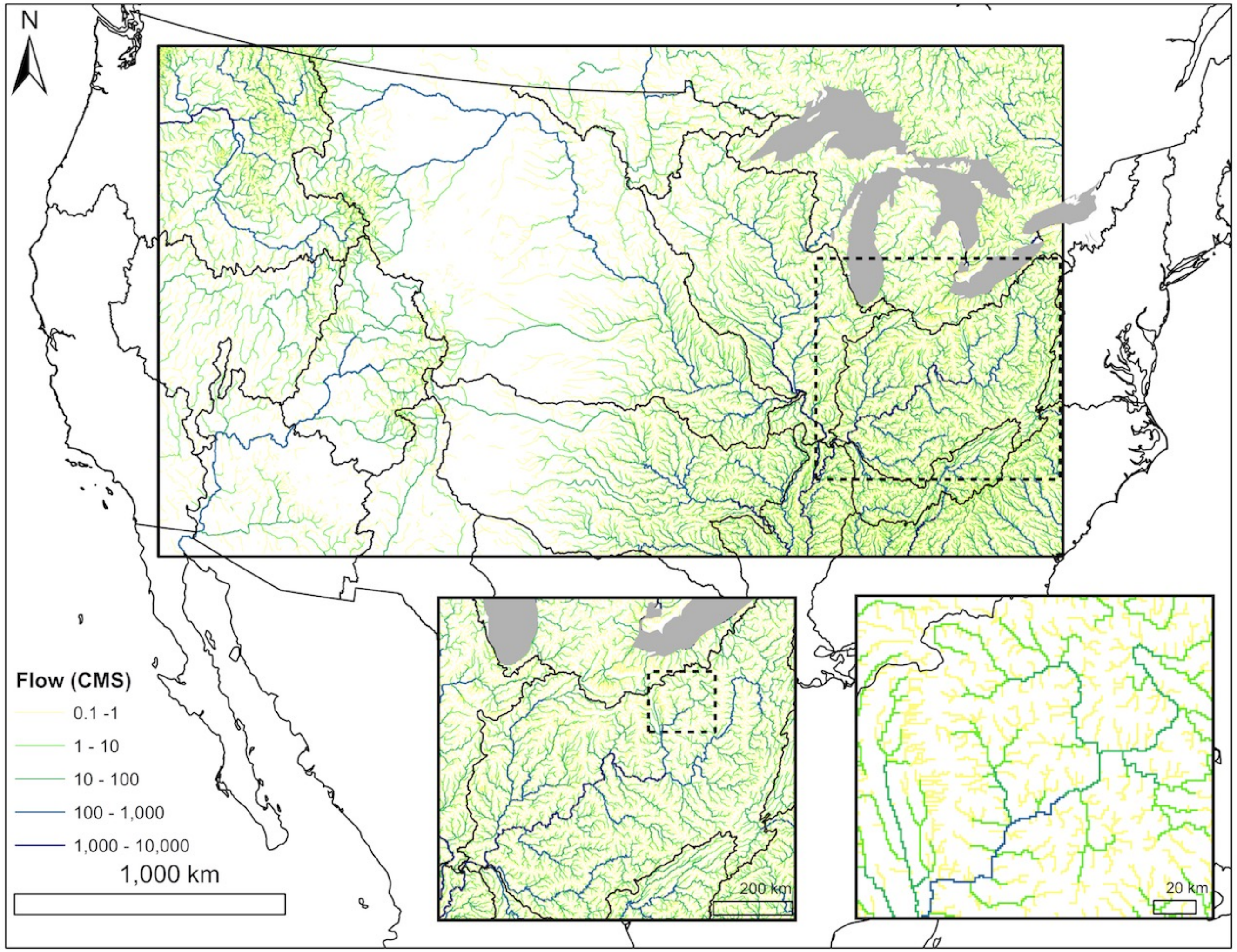
In a hypothetical battle between increased ET
and phase change, ET wins



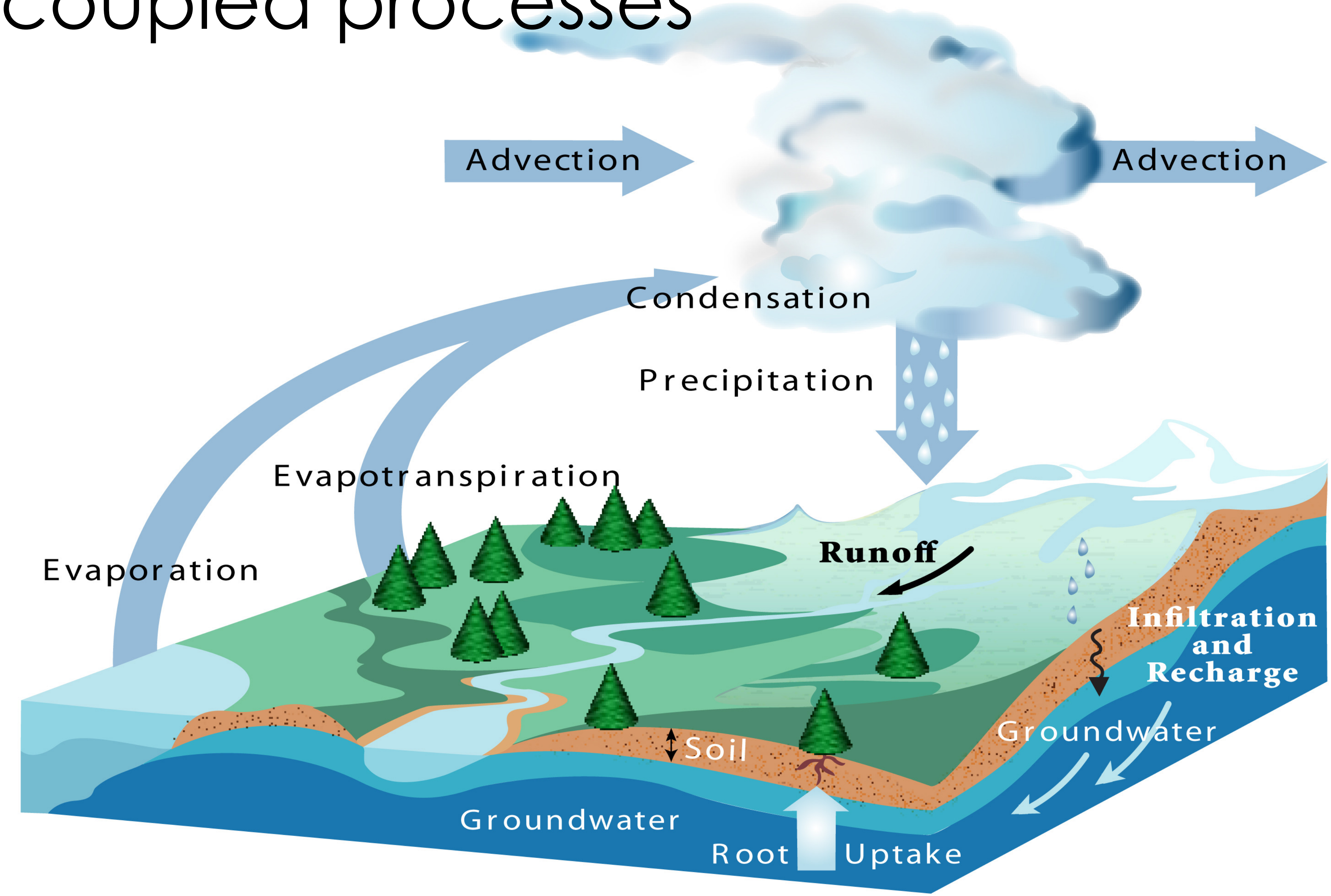
Foster et al *ERL* 2016

Many things pretend not to be models but really are

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*

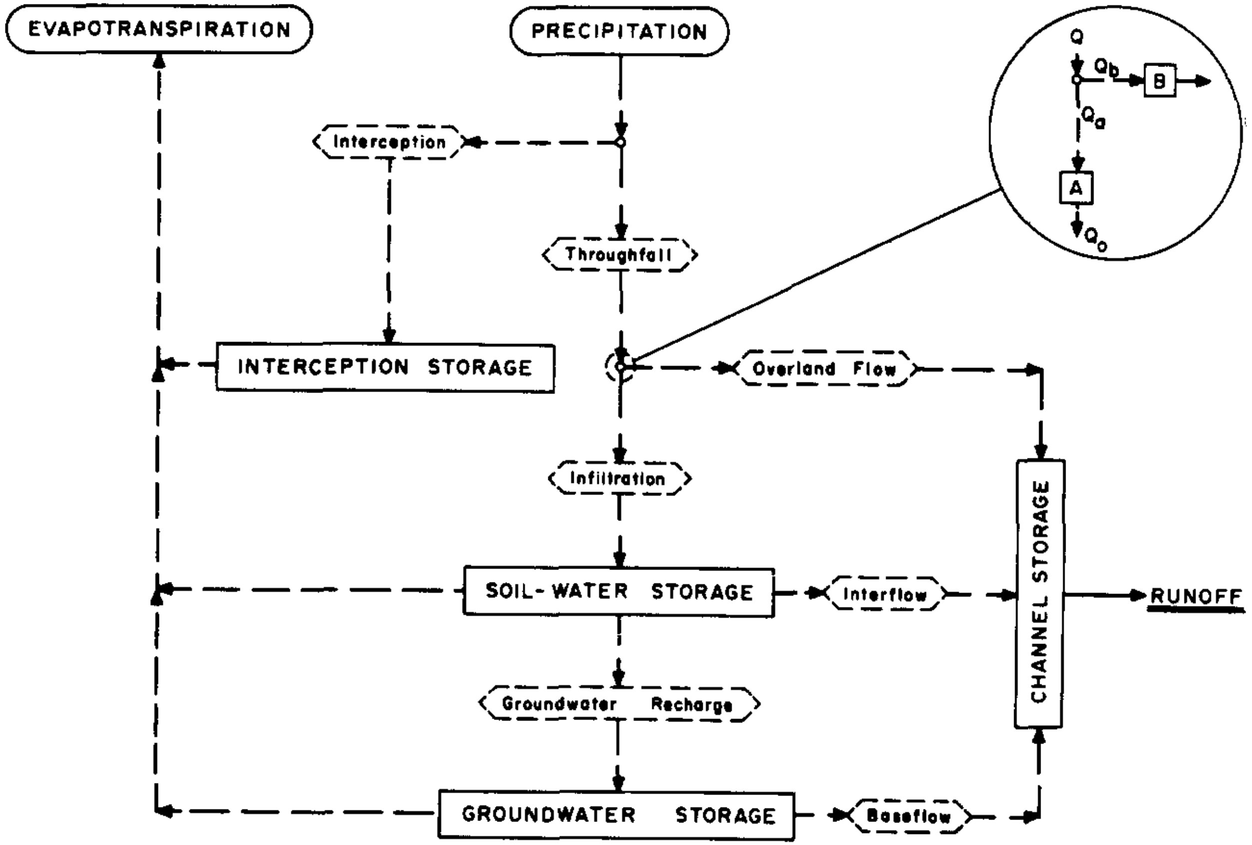
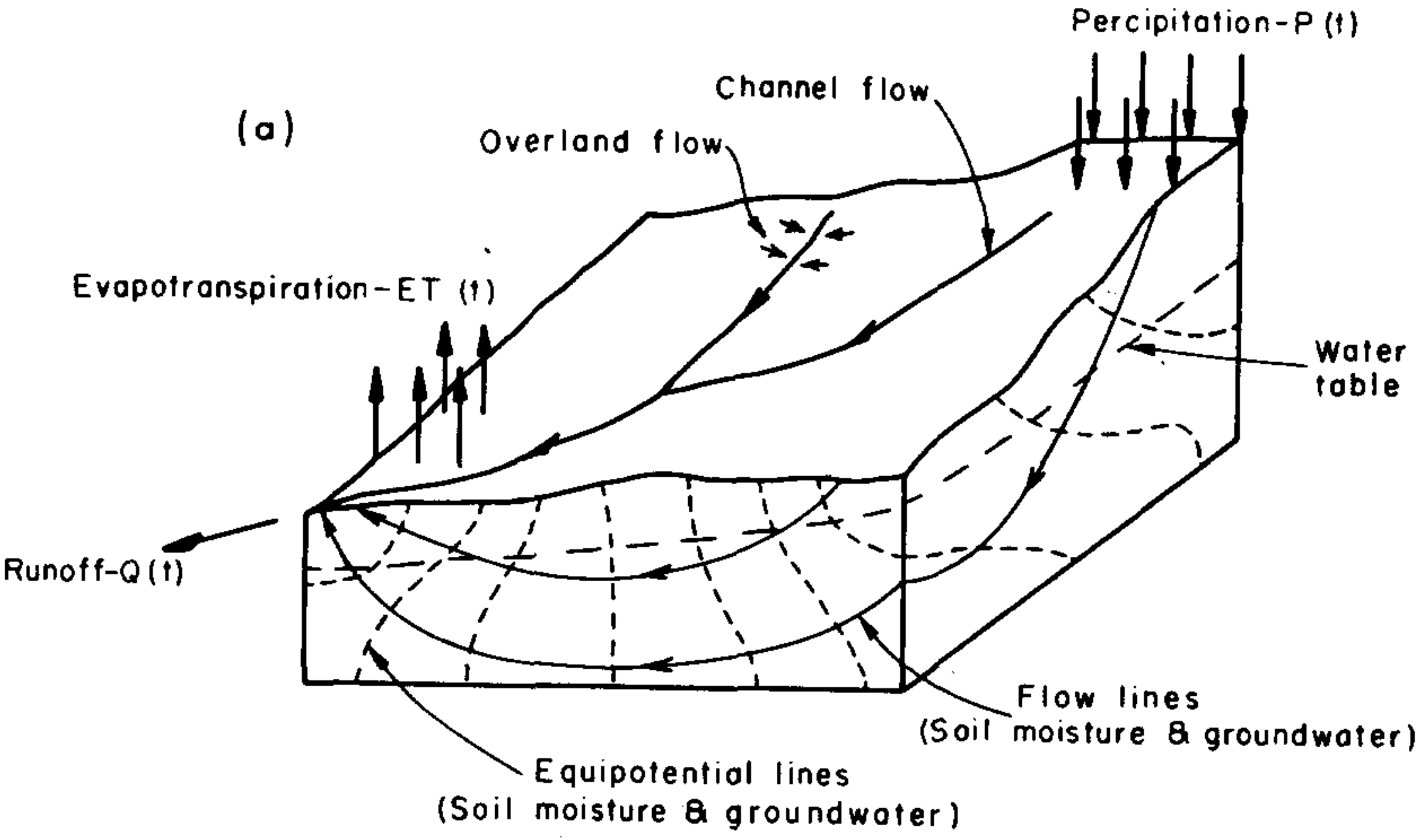


Fig. 2. A conceptual hydrologic model of the type used in the stepwise routing approach of systems hydrology.

Freeze and Harlan (1969)

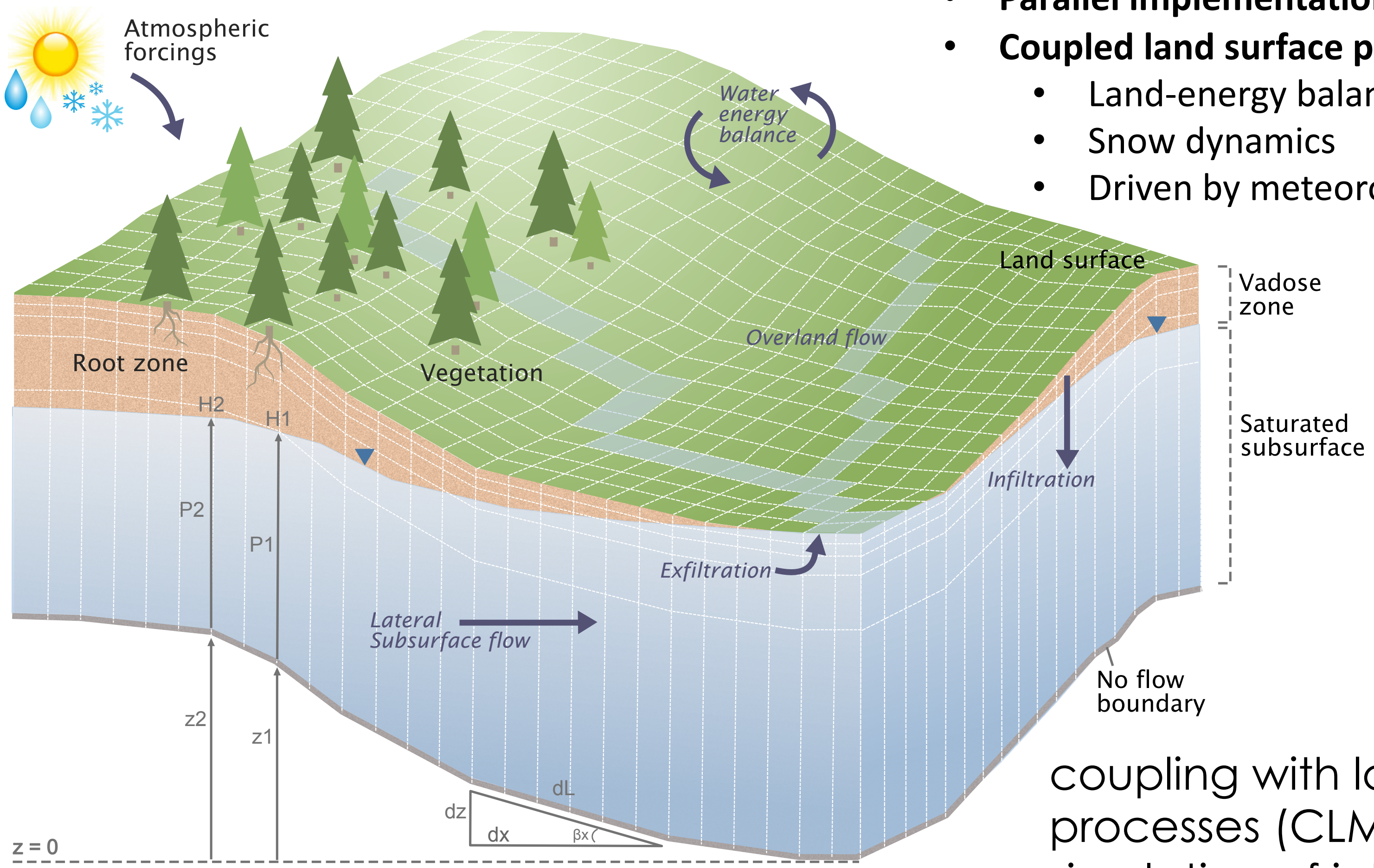
They introduced a conceptual model that better represented the way we still envision hydrologic systems



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)

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

ParFlow Equations: Richards' and Overland Flow

$$S_s S_w \frac{\partial \psi_p}{\partial t} + \phi \frac{\partial S_w(\psi_p)}{\partial t} = \nabla \cdot \mathrm{q} + q_s$$

$$\boldsymbol{q} = -k(x) k_r(\psi_p) \nabla(\psi_p - z)$$

Richards' Equation

Pressure Head (L)

Saturation (-)

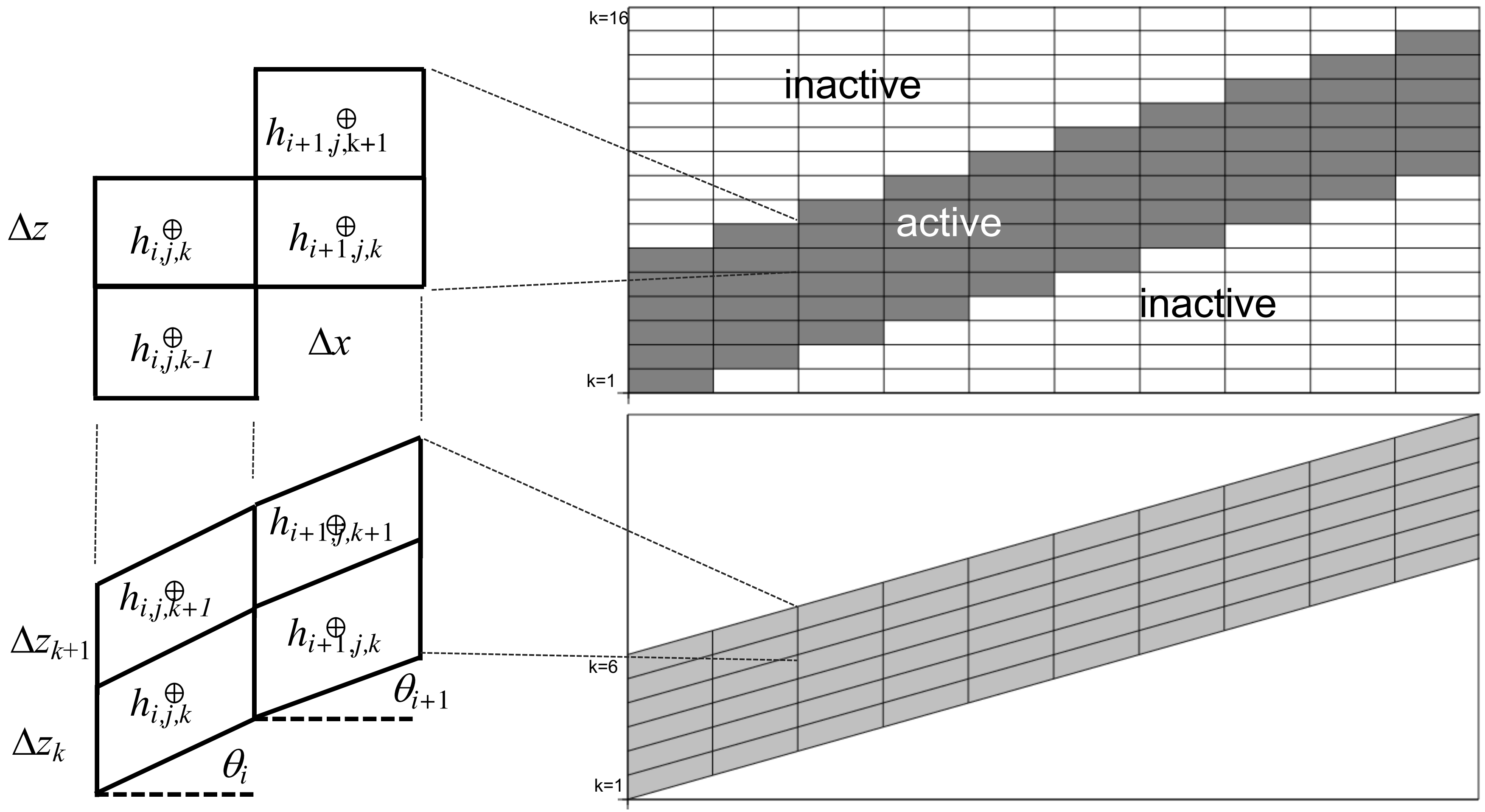
Saturated Hydraulic Conductivity (L/T)

$$\frac{\partial \psi_s}{\partial t} = \nabla \vec{\mathrm{v}} \psi_s + q_r$$

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

Kinematic Wave Equation

ParFlow Gridding



Terrain Following Grid EQ

Modified Darcy's Law:

$$\mathbf{q} = -\mathbf{K}_s(\mathbf{x})k_r(h)[\nabla(h+z)\cos\theta_x + \sin\theta_x]$$

Slopes and fluxes:

$$\theta_x = \tan^{-1}(S_{0,x}) \text{ and } \theta_y = \tan^{-1}(S_{0,y})$$

$$q_x = -K_{s,x}(\mathbf{x})k_r(h)\left[\frac{\partial(h)}{\partial x}\cos\theta_x + \sin\theta_x\right]$$

$$= -K_{s,x}(x)k_r(h)\frac{\partial(h)}{\partial x}\cos\theta_x - K_{s,x}(x)k_r(h)\sin\theta_x$$

Diffusive Pressure Term Topographic Term

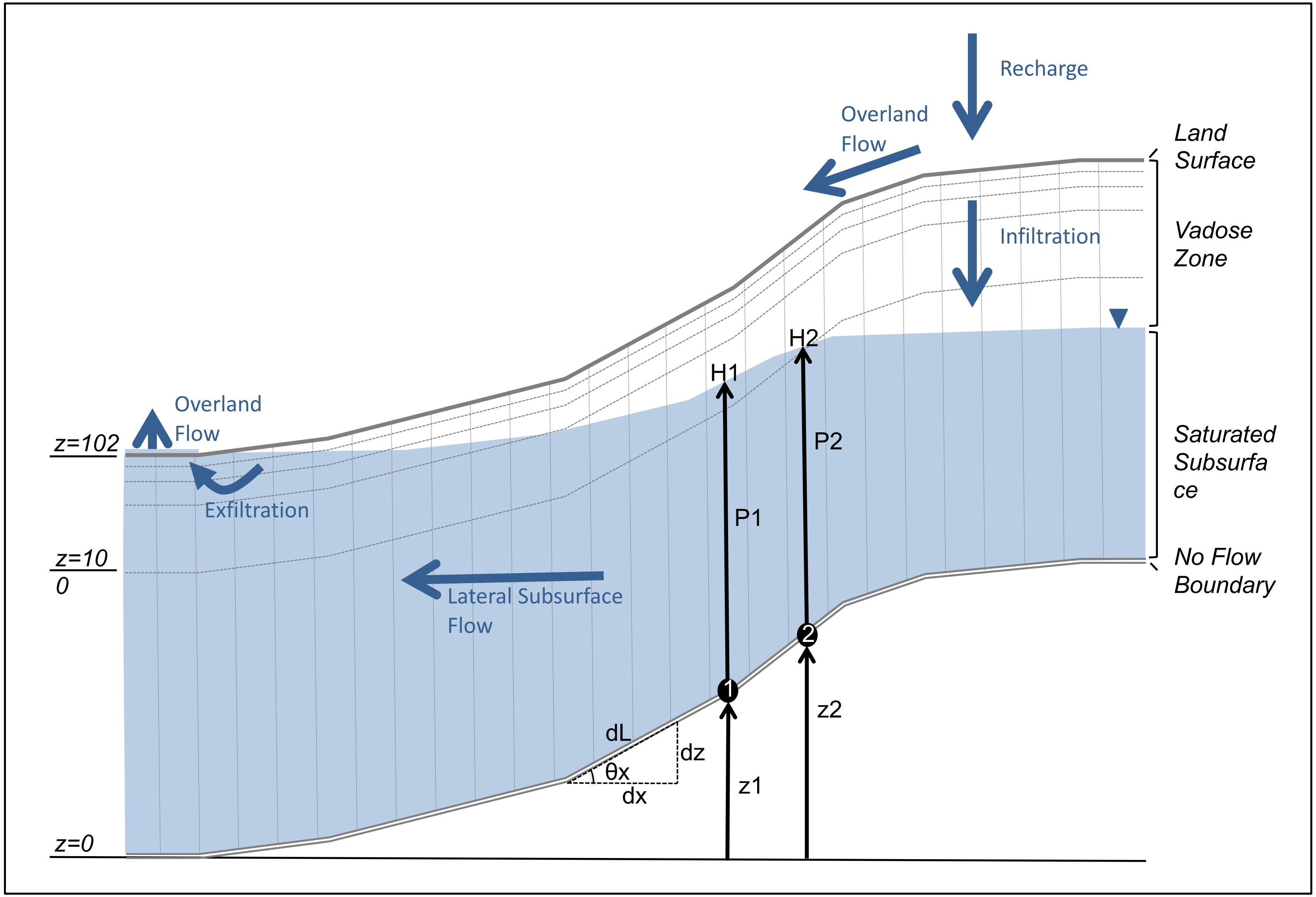
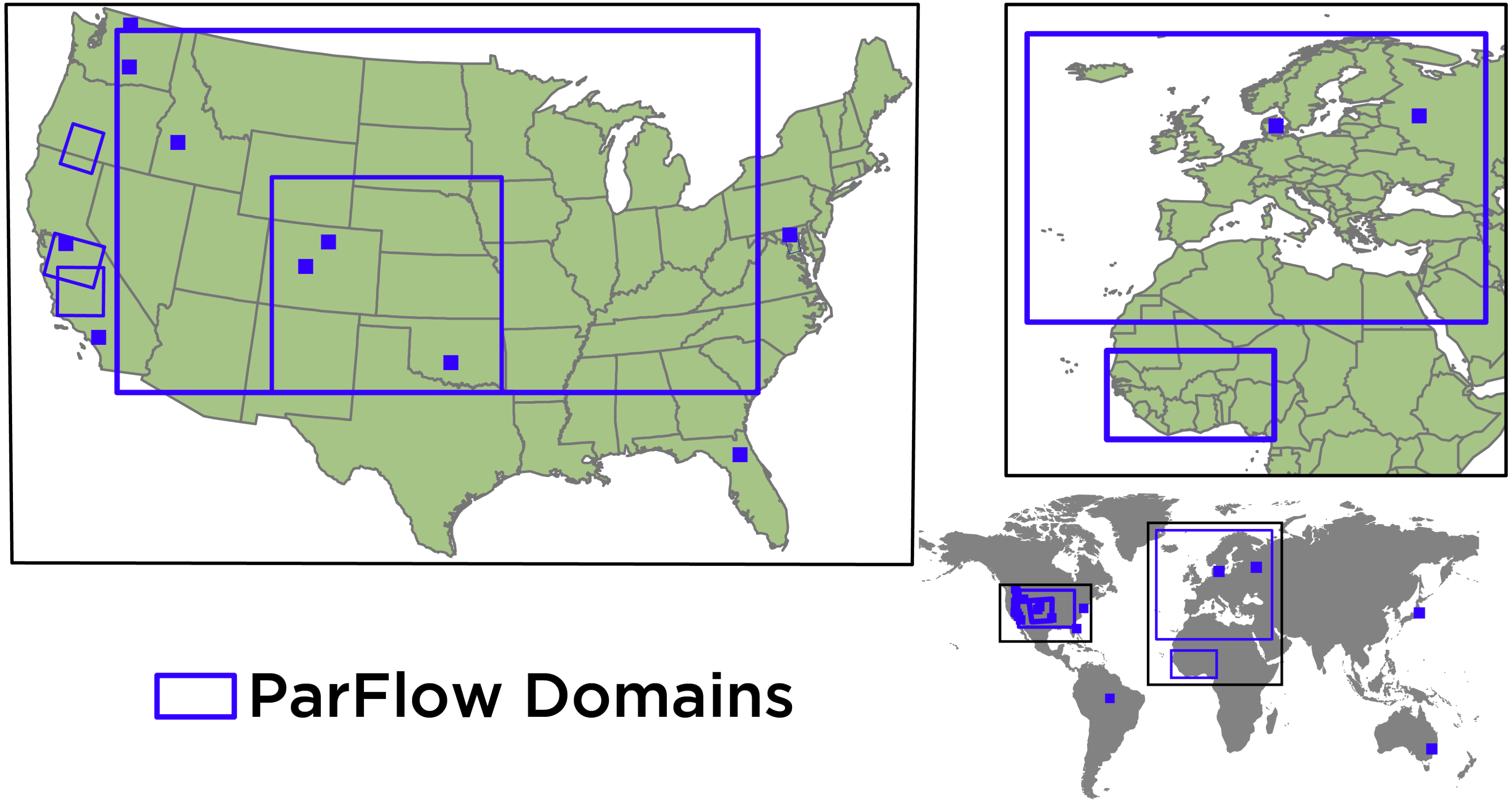
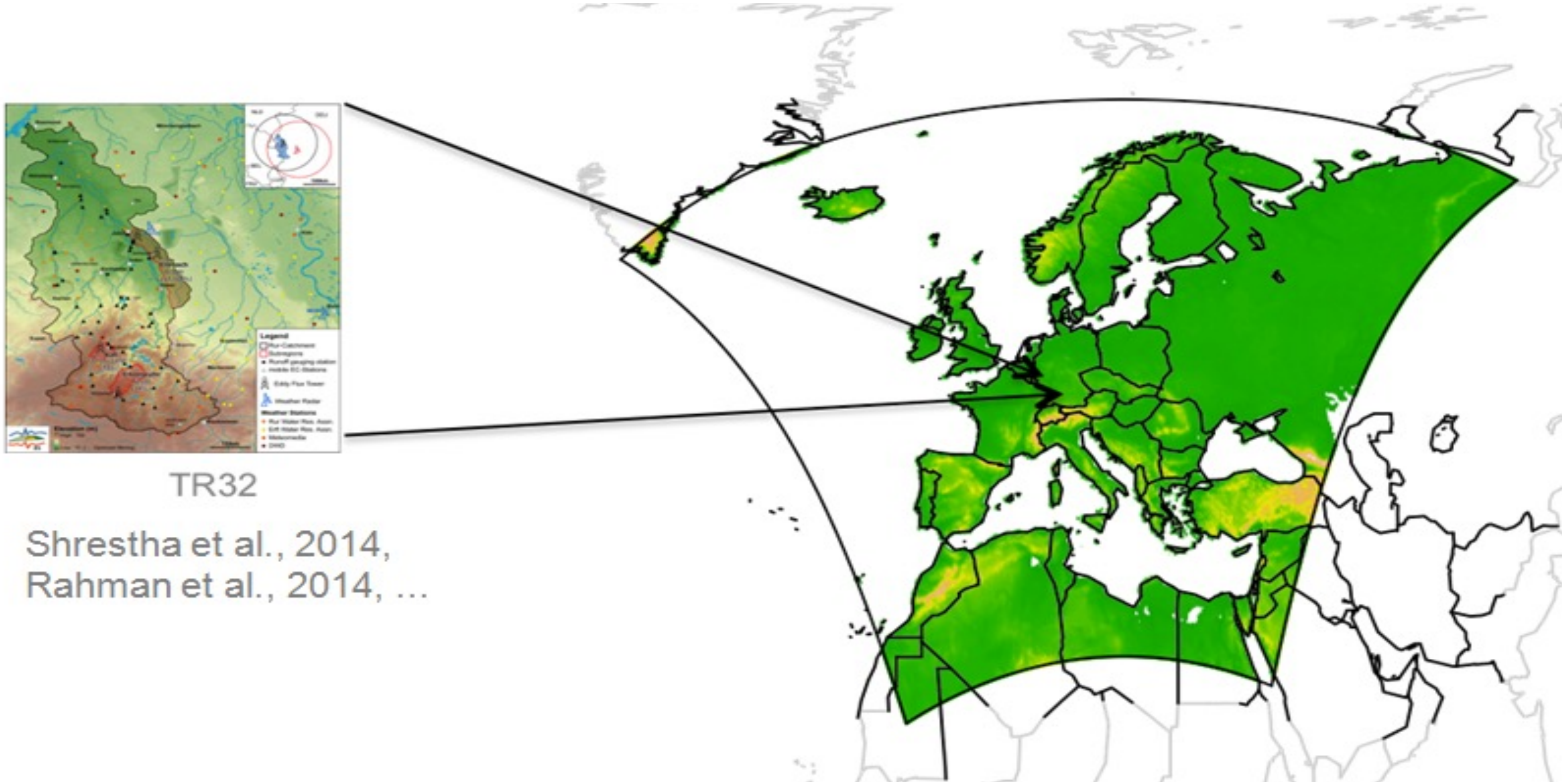


Illustration of a ParFlow model for an idealized hillslope using the terrain following grid formulation

ParFlow has been applied at many sites worldwide



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



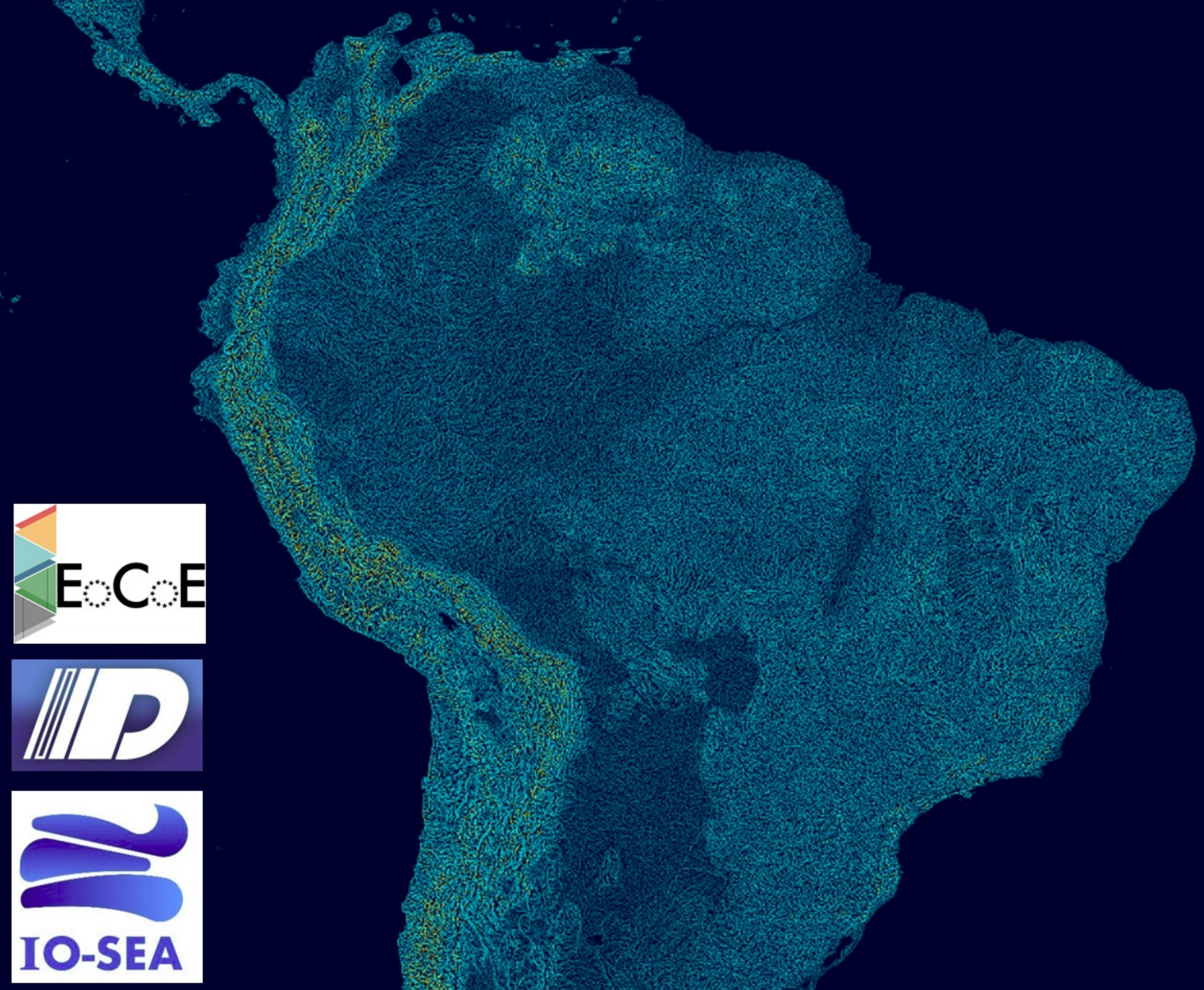
TR32

Shrestha et al., 2014,
Rahman et al., 2014, ...

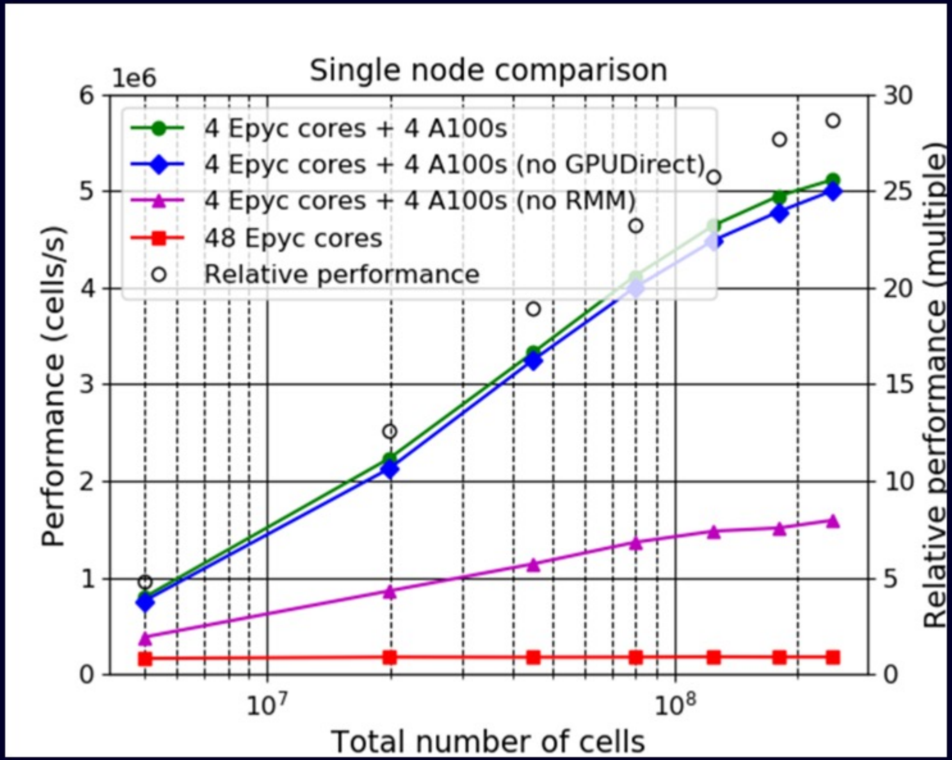
Including Big Thompson (CO), Chesapeake (MD), CONUS, CORDEX, East Inlet (CO), East River (CO), Klamath (OR), Little Washita (OK), Rur (Germany), San Joaquin (CA), Sante Fe (FL), Skjern (Denmark), Wüstebach (Germany)

EXA-SCALE READY GLOBAL HYDROLOGIC SIMULATIONS

ParFlow-GPU, JUWELS-Booster, global, 1km resolution, here: S. America, surface soil moisture



- 320 NVIDIA A100 GPUSs distributed over 80 compute nodes
- Output per time step: 60GB per variable

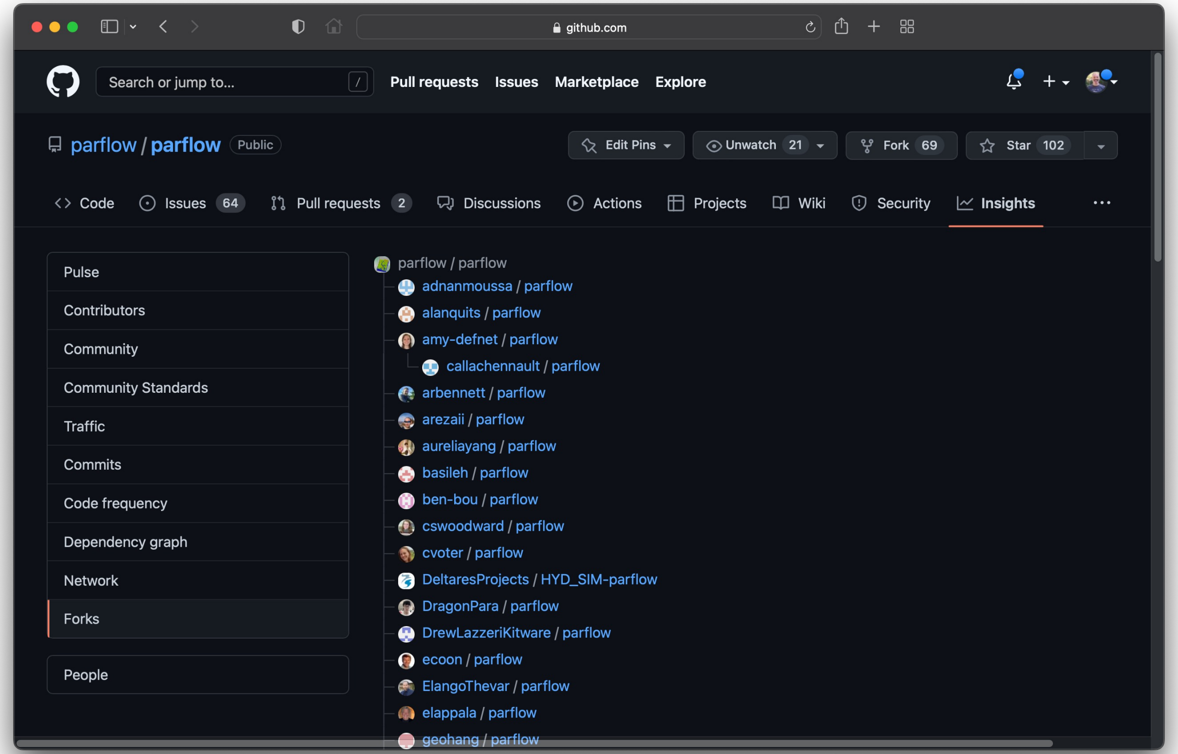


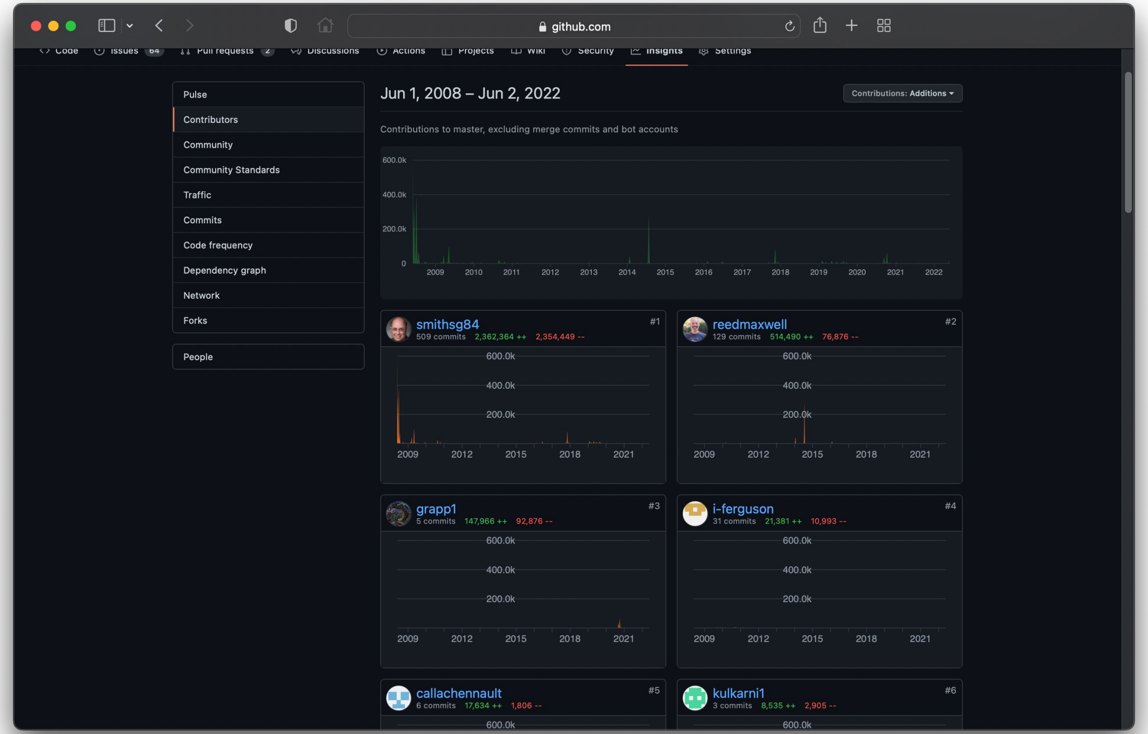
Hokkanen et al., 2021

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

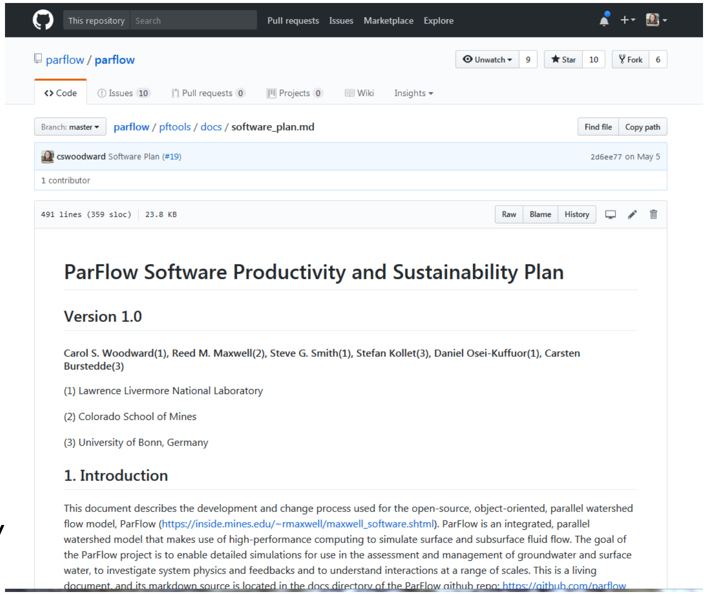
ParFlow is a community code, developed by several groups





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



ParFlow Software Productivity and Sustainability Plan

Version 1.0

Carol S. Woodward(1), Reed M. Maxwell(2), Steve G. Smith(1), Stefan Kollet(3), Daniel Osei-Kuffuor(1), Carsten Burstedde(3)

(1) Lawrence Livermore National Laboratory

(2) Colorado School of Mines

(3) University of Bonn, Germany

1. Introduction

This document describes the development and change process used for the open-source, object-oriented, parallel watershed flow model, ParFlow (https://inside.mines.edu/~rmaxwell/maxwell_software.shtml). ParFlow is an integrated, parallel watershed model that makes use of high-performance computing to simulate surface and subsurface fluid flow. The goal of the ParFlow project is to enable detailed simulations for use in the assessment and management of groundwater and surface water, to investigate system physics and feedbacks and to understand interactions at a range of scales. This is a living document, and its markdown source is located in the docs directory of the ParFlow github repo: https://github.com/parflow

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

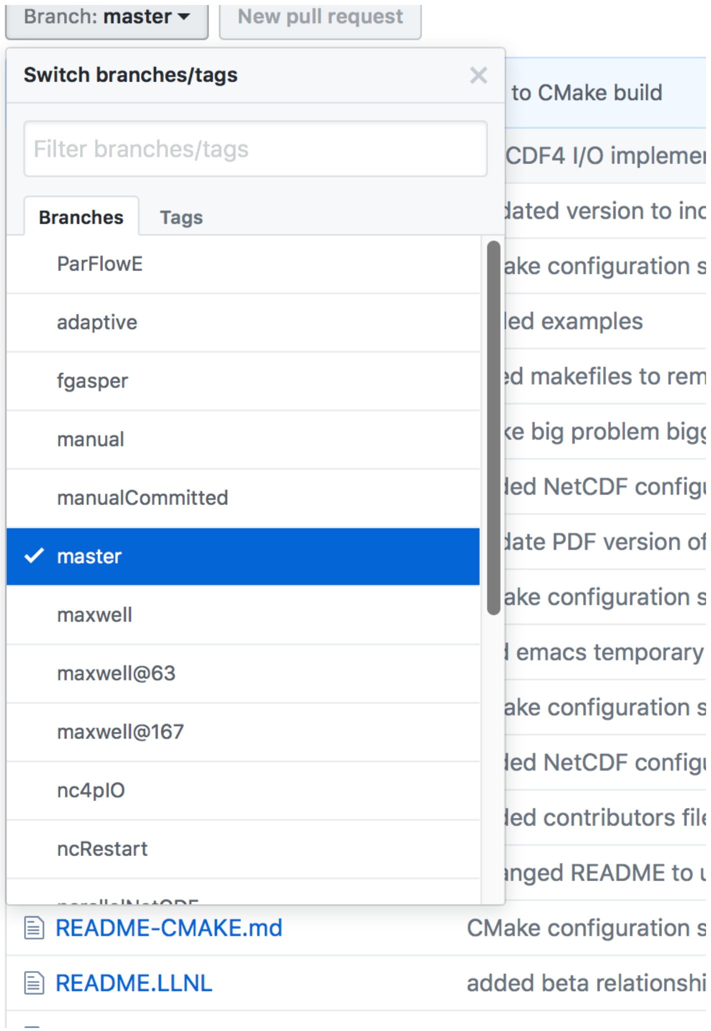
Centralized version control and development is critical

- 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

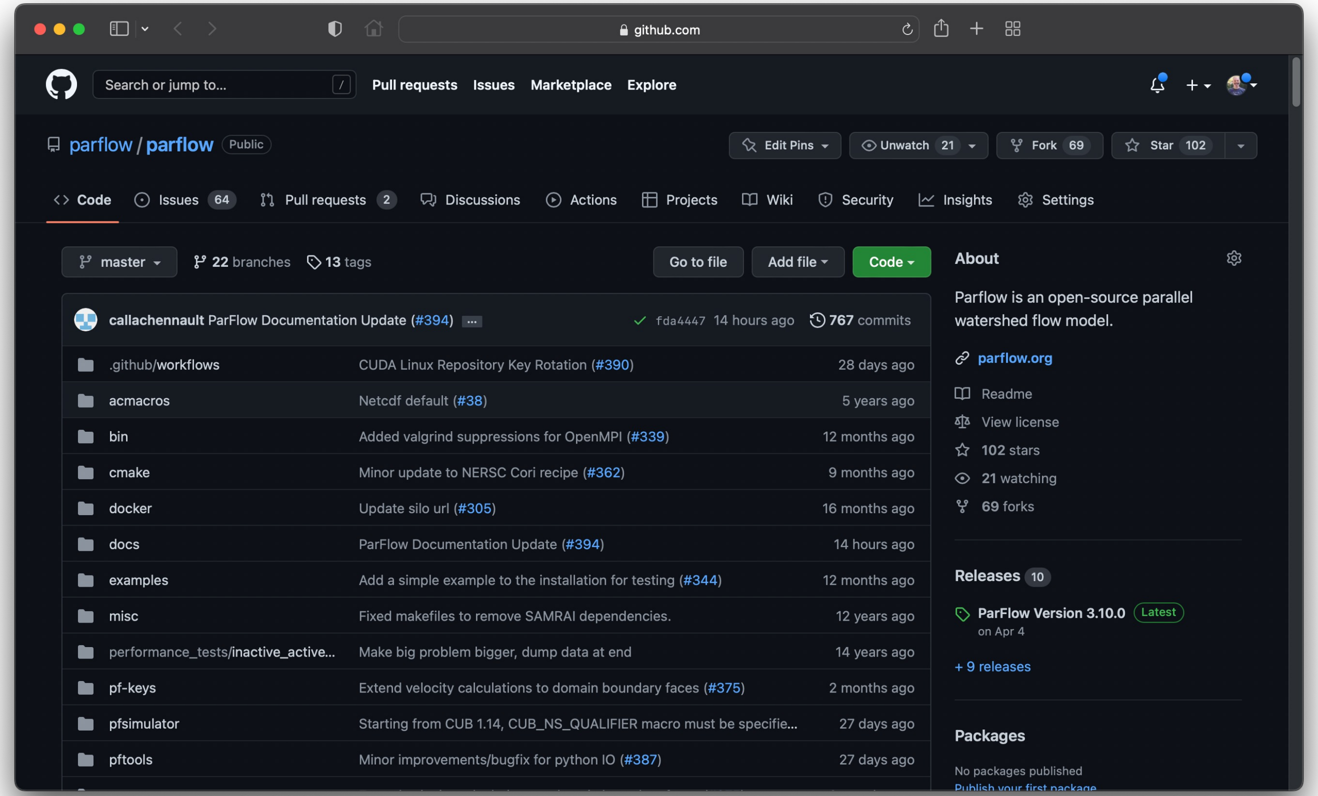
With many active development branches the development plan is important

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



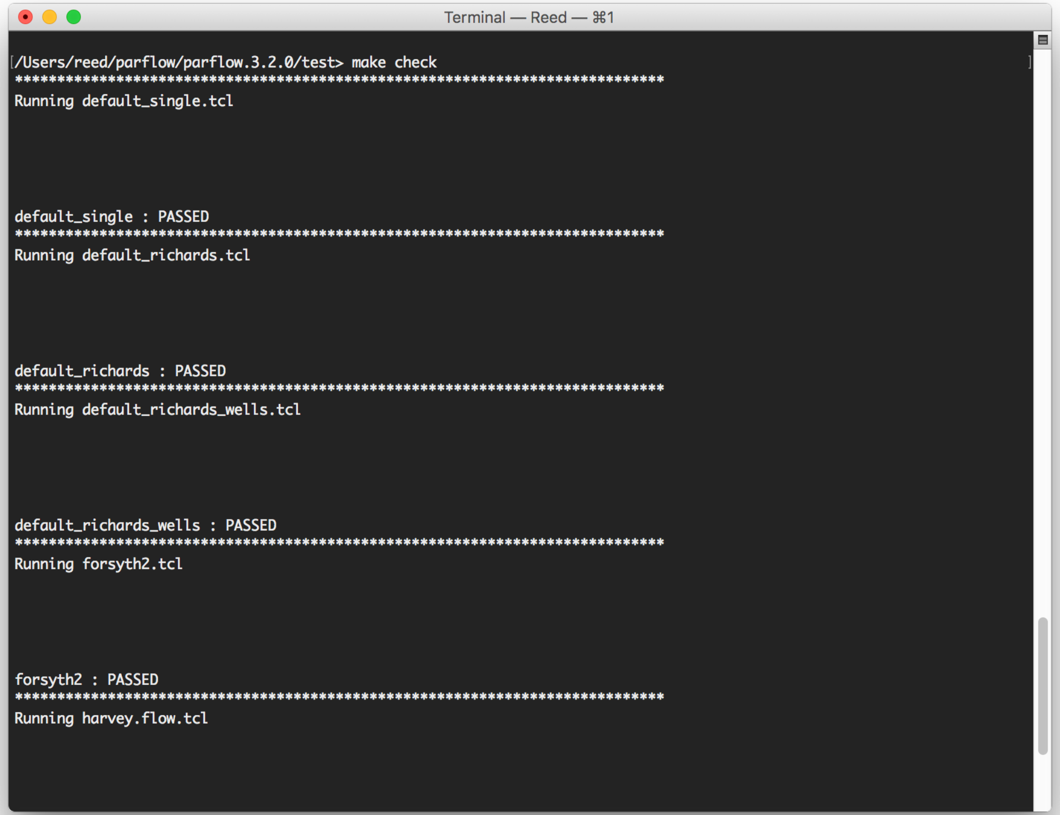
GitHub provides a mechanism to enforce the development plan

- 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



We last had a short course in 2019

Since that time many things have evolved in ParFlow

- Full Docker capability
- Python Interface
- CUDA, OpenMP, KOKKOS back-ends
- RST Manual
- Integration with ParaView and other vis platforms
- 10x more developers
- 50x more users

Course set up

- Hybrid mix of TCL and Python
- Local builds
- Very free structured

Take Home Messages…

- We can strive towards an integrated picture, model and understanding of the hydrologic cycle
- This requires new equations, process descriptions, solvers and parallel architecture
- This enables new understanding about connections between components

The hydrostatic profile