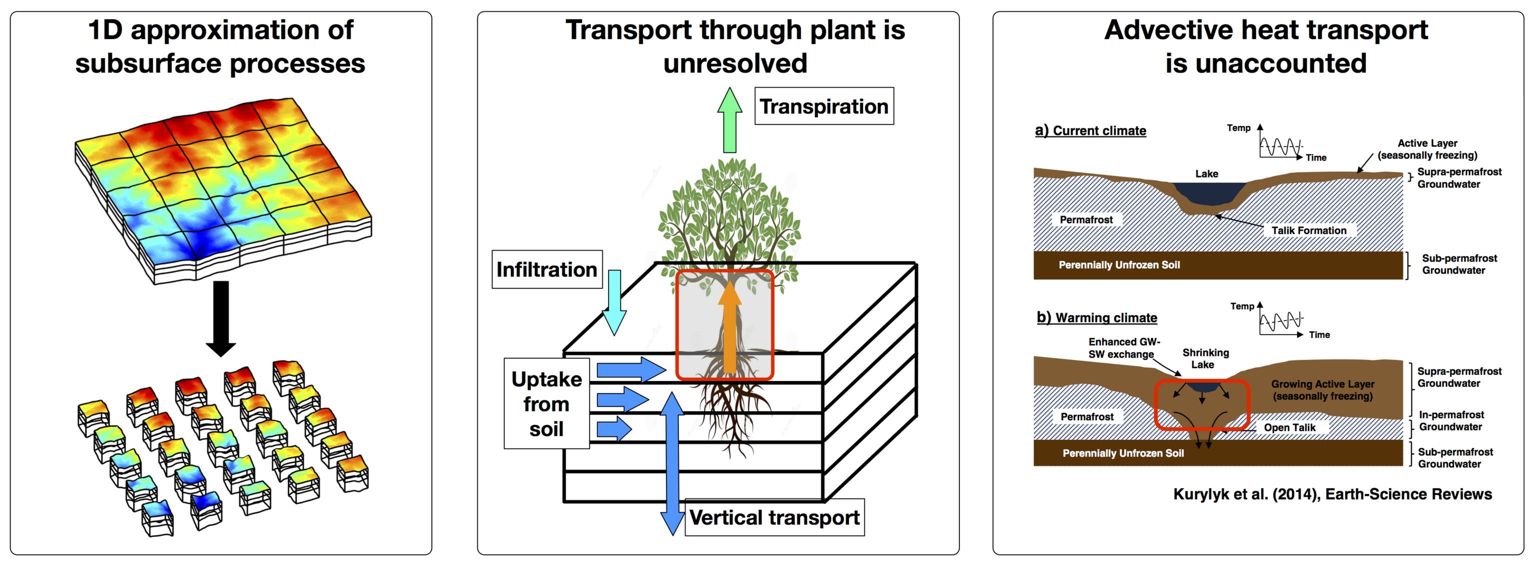
Development of Terrestrial Dynamical Core for the E3SM to Simulate Water Cycle

Gautam Bisht[1] (gautam.bisht@pnnl.gov), Jed Brown[2], Nathan Collier[3], Jennifer Frederick[4], Glenn Hammond[4], Satish Karra[5], and Mathew Knepley[6]

[1]PNNL, [2]UC-Boulder, [3]ORNL, [4]SNL, [5]LANL, [6]U. Buffalo

Motivation

- Developing a predictive understanding of the terrestrial water cycle at local to global scale is essential for accurate assessment of water resources, agricultural production, and energy generation given current climate variability
- Terrestrial component of the DOE's E3SM excludes many critical physical processes:



Objective

- Develop a rigorously verified, spatially adaptive, scalable, multi-physics dynamical core for global-scale modeling of three-dimensional subsurface processes for E3SM

Terrestrial Dynamical core

The **T**errestrial **Dy**namical **core** (TDycore) will solve 3D transport of water and energy in the subsurface given by:

$$\frac{\partial}{\partial t}(\rho\phi s) = -\nabla\cdot(\rho\mathbf{q}) + Q_w \quad (1)$$

$$\frac{\partial}{\partial t}(\rho\phi s U + (1-\phi)\rho_p C_p T) = -\nabla\cdot(\rho\mathbf{q}H - \kappa\nabla T) + Q_e \quad (2)$$

where $\mathbf{q} = \frac{k_r K}{\mu}\nabla(P + \rho g z)$

We are pursuing a two pronged development that is focused on:

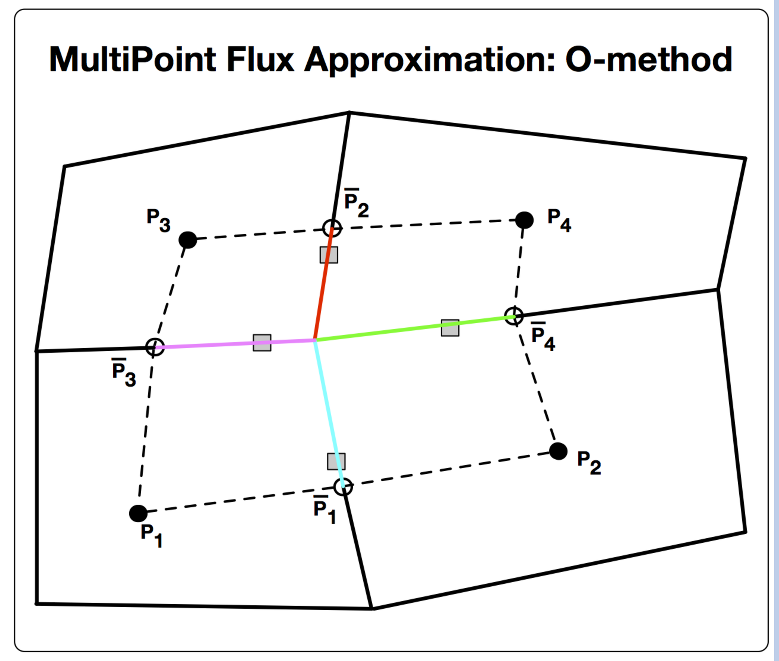
1. Using spatial discretization methods that accounts for non-orthogonal grids
2. Using a flexible framework that supports with different temporal discretization schemes

Development of TDycore library

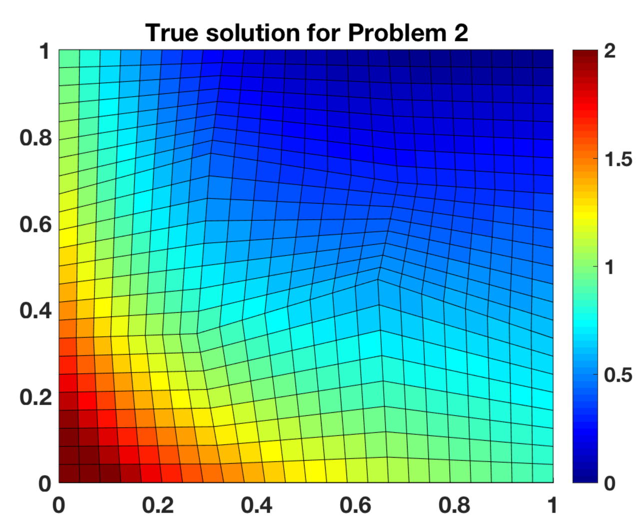
- Developing a scalable *library* on top of PETSc framework
- E3SM terrestrial dycore will be an application of the TDycore library
- Open-source and open-development
- Core library is written **in C with Fortran bindings**
- Supports runtime configurability: `-tdy_method {wy|mpfao|...}`
- Using a modified PFLOTRAN's regression testing framework
- Includes 5 demo applications and 27 regression tests
- Available on Github: `https://github.com/TDycores-Project/TDycore`
- Using Travis-CI for regression testing `https://travis-ci.org/TDycores-Project/TDycore`
- Regression tests cover 95% of the code `https://codecov.io/gh/TDycores-Project/TDycore`

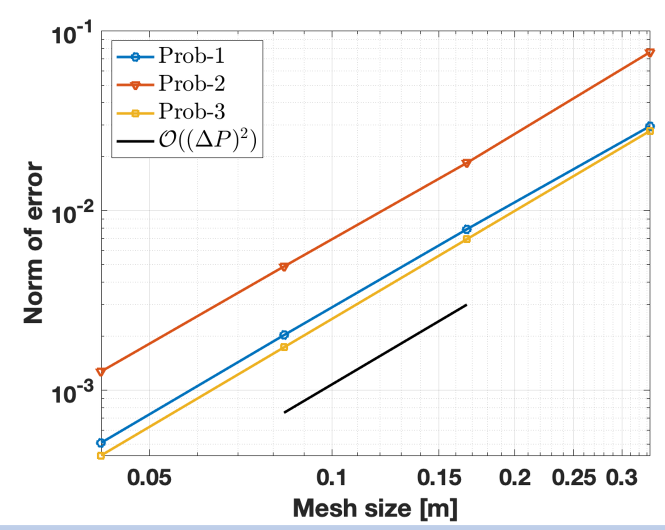
Multi Point Flux Approximation: O-method

- Based on finite volume in which control volumes are subdivided into interaction volumes (IVs)
- Pressure varies linearly and flux continuity is enforced across IVs
- Discretization can be performed in physical or reference space
- Number of unknowns are cell-centered pressure values
- Method of Manufactured Solutions is used to verify the serial implementation

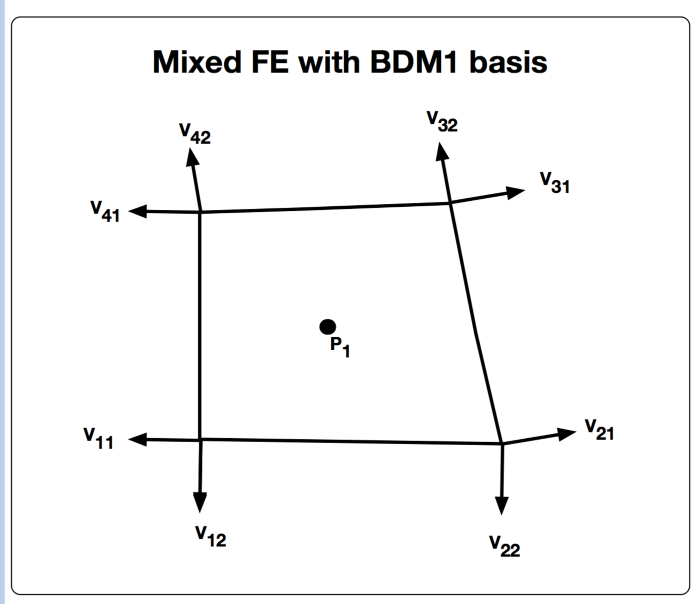


1. $P = 3.14 + x(1-x) + y(1-y)$ and $K_{11} = 5$; $K_{22} = 2$; $K_{12} = K_{21} = 1$
2. $P = (x-1)^4 + (1-x)(1-y)^3 + \sin(1-y)\cos(1-x)$ and $K_{11} = 5$; $K_{22} = 2$; $K_{12} = K_{21} = 1$
3. $P = x(1-x) + y(1-y) + z(1-z)$ and $K_{11} = 5$; $K_{22} = 2$; $K_{33} = 3$ and $K_{ij} = 1$ for $i \neq j$





Mixed Finite Element

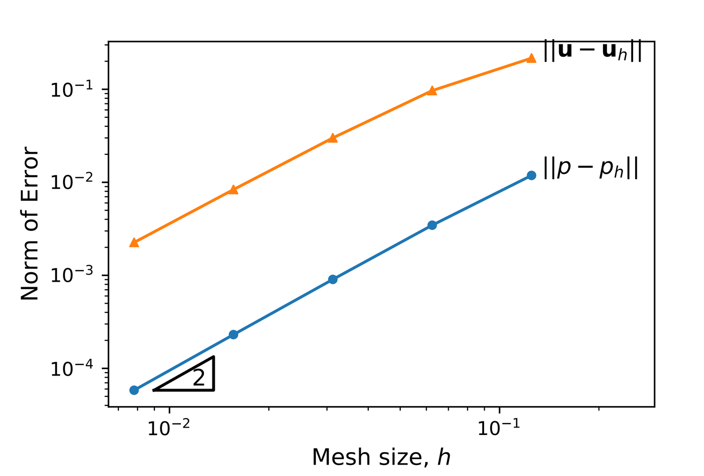


- Handles discontinuous coefficients on non-orthogonal grids
- BDM1 basis assumes that normal velocity may vary linearly along a edge and pressure is constant
- Leads to a saddle point problem that requires specialized preconditioning techniques
- Also implemented Wheeler-Yotov (WY) method,

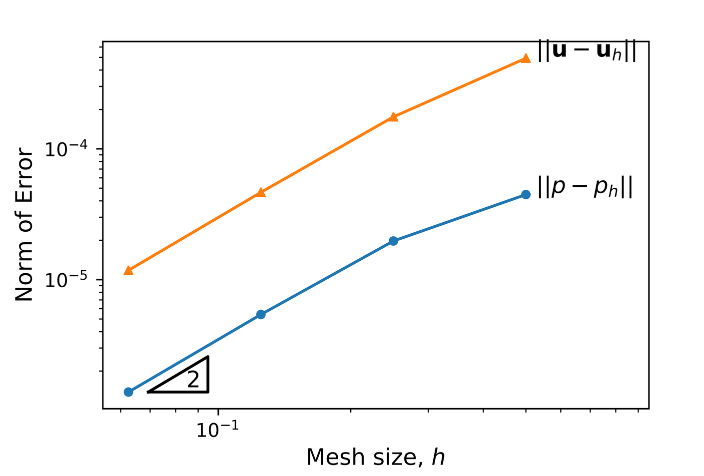
which uses special quadrature to allow for local velocity elimination

- WY leads to a symmetric and positive definite cell-centered system for pressure unknowns
- Implemented and tested for distorted 2D and 3D grids in parallel

Problem 2

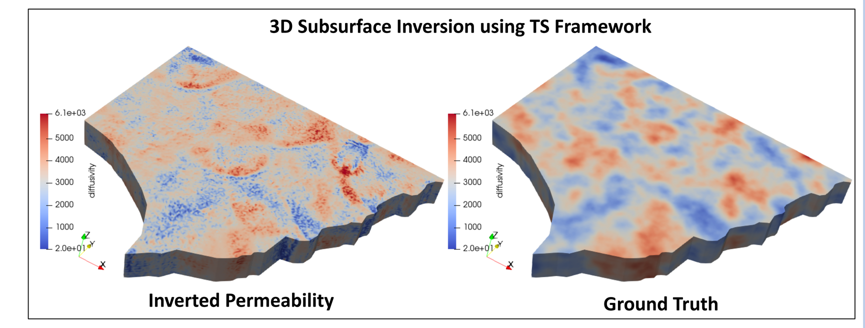


Problem 3



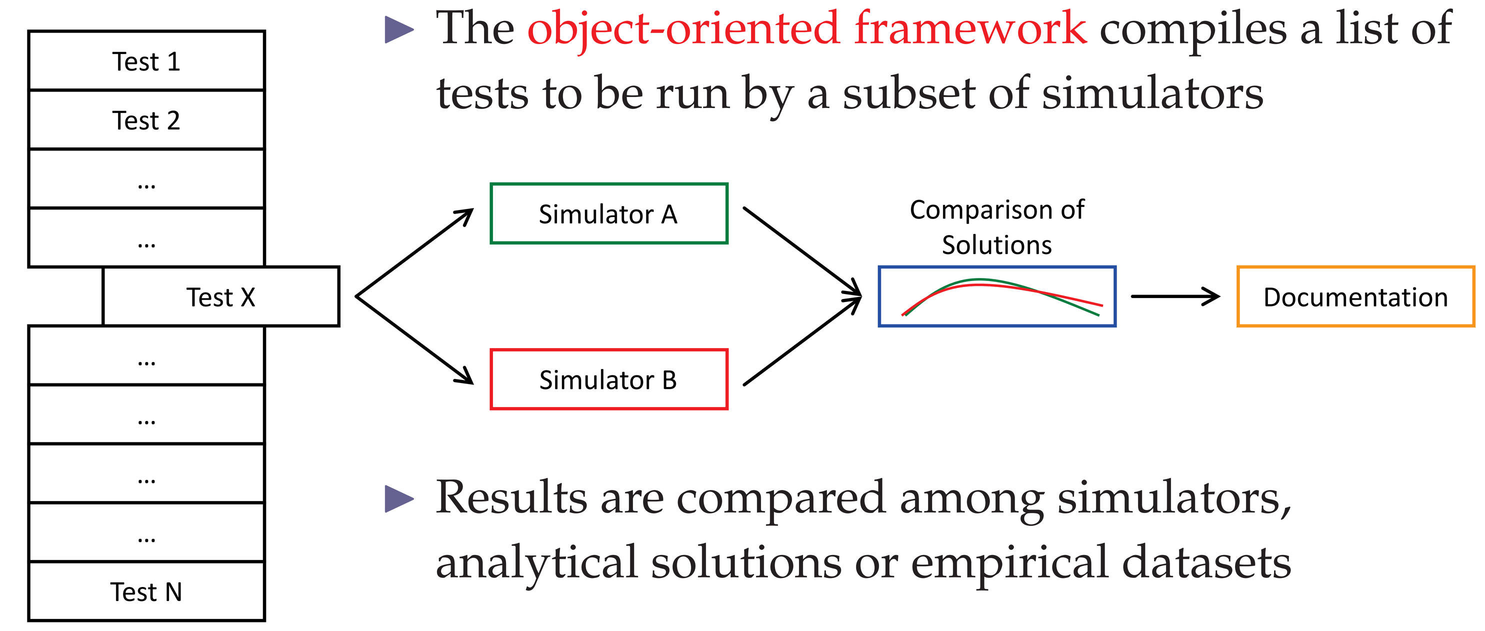
Temporal discretization: Methods

- Implemented and tested flexible temporal discretization framework, PETSc TS, for the mass and energy subsurface governing equations
- Allows to choose from 12 types of time integrators (e.g., Backward Euler, Crank-Nicholson, Theta, etc.) on the fly without any new code development
- TS framework can be used to build discrete adjoints for sensitivity analysis and inversion with minimal code addition

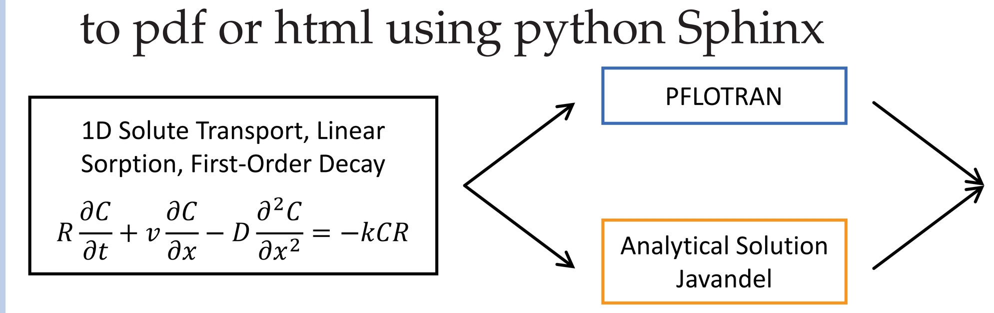


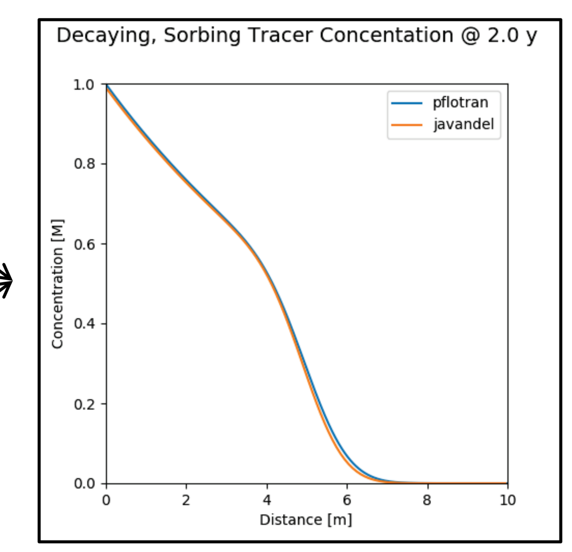
V&V framework

- Developing an automated, python-based framework for V&V testing in the cloud that is code-agnostic



- The object-oriented framework compiles a list of tests to be run by a subset of simulators
- Results are compared among simulators, analytical solutions or empirical datasets
- Documentation is generated in reStructuredText format and compiled to pdf or html using python Sphinx





Next steps

- Extend MPFA-O method to support multiple processors
- Perform an inter-comparison of spatial discretization methods
- Combine the developments in spatial and temporal discretization methods to solve a non-linear, transient subsurface flow problem on non-orthogonal grids
- Use the V&V framework to benchmark the dycore
- Couple the TDycore with ELM for watershed scale simulation

Acknowledgements

This work is supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, and Office of Advanced Scientific Computing Research Scientific Discovery through Advanced Computing (SciDAC) program