Development of Terrestrial Dynamical Cores for E3SM

Gautam Bisht[1], Jed Brown[2], Nathan Collier[3], Jennifer Fredrick[4], Glenn Hammond[4], Satish Karra[5], and Mathew Knepley[6]

2019 Scientific Discovery through Advanced Computing Principal Investigator (PI) Meeting

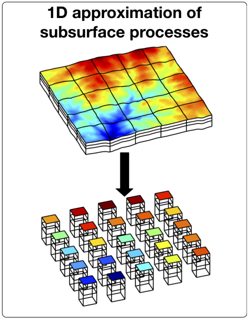
July 17, 2019

[1]PNNL, [2]Univ. Colorado, Boulder, [3]ORNL, [4]SNL, [5]LANL, [6]Univ. Buffalo

PNNL is operated by Battelle for the U.S. Department of Energy

Land surface models need to move beyond 1D models

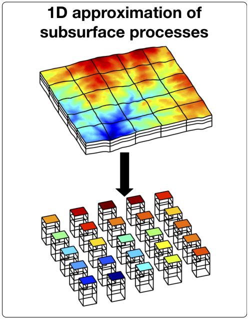
- Fan et al. (2019) identified including 3D subsurface flow as a key process for next-generation LSMs



Land surface models need to move beyond 1D models

- Fan et al. (2019) identified including 3D subsurface flow as a key process for next-generation LSMs

- Lateral redistribution of soil moisture leads to an increase in predicted surface fluxes at watershed (Tague and Peng, 2013) and continental scales (Maxwell and Condon, 2016)

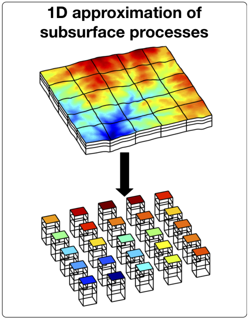


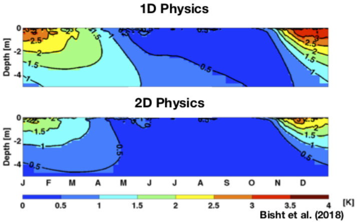
Land surface models need to move beyond 1D models

- Fan et al. (2019) identified including 3D subsurface flow as a key process for next-generation LSMs

- Lateral redistribution of soil moisture leads to an increase in predicted surface fluxes at watershed (Tague and Peng, 2013) and continental scales (Maxwell and Condon, 2016)

- Exclusion of lateral redistribution of subsurface heat leads to an overestimation of spatial variability in soil temperature (Bisht et al., 2018)





Spatial std. dev. of T_{soil} for a transect in Barrow, AK

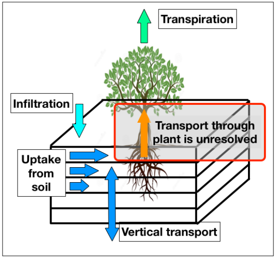
Need for higher fidelity in LSMs

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

Need for higher fidelity in LSMs

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

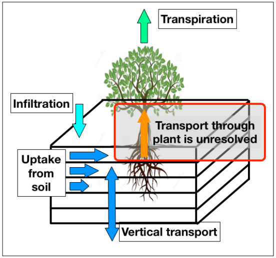
- Transport of water through soil-plant continuum



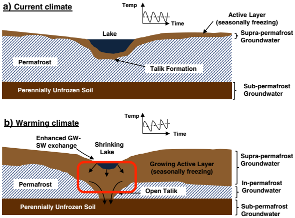
Need for higher fidelity in LSMs

Current generation land surface models (LSMs), including ELM, routinely neglect many critical multi-component, multi-physics processes such as:

- Transport of water through soil-plant continuum



- Advective transport of energy

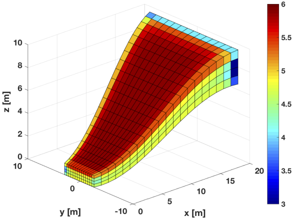


Kurylyk et al. (2014), Earth-Science Reviews

Computational challenges for a 3D global LSM

E3SM's 10-year vision of a sub-kilometer resolution in terrestrial components imposes several key computation requirements for the terrestrial dynamical core (dycore):

- Scalable solver for nonlinear parabolic PDE with 10^{10} unknowns
- Spatial discretization that accounts for non-orthogonal unstructured grids
- Flexible framework to assemble a tightly coupled multi-component, multi-physics problem
- Runtime configurability to use a range of numerical algorithms



Goal

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

The coupled thermal-hydrology model

The terrestrial dycore will solve 3D transport of water and energy in the subsurface given by:

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

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

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

The coupled thermal-hydrology model

The terrestrial dycore will solve 3D transport of water and energy in the subsurface given by:

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

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

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

The coupled thermal-hydrology model

The terrestrial dycore 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 experimenting with different temporal discretization schemes

Terrestrial **Dy**namical **core** (**TDycore**) library

- Developing a scalable **library** on top of PETSc framework

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM dycore will be an **application** of the TDycore library

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM dycore will be an **application** of the TDycore library
- Open-source and open-development

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM dycore will be an **application** of the TDycore library
- Open-source and open-development
- Core library is written in C with Fortran bindings

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM 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|...`}

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM 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|...`}
- Adopted PFLOTRAN's regression testing framework for the TDycore lib

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM 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|...}`
- Adopted PFLOTRAN's regression testing framework for the TDycore lib
- Includes 5 demo applications and 27 regression tests

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM 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|...}`
- Adopted PFLOTRAN's regression testing framework for the TDycore lib
- Includes 5 demo applications and 27 regression tests
- Available at `https://github.com/TDycores-Project/TDycore`

Terrestrial Dynamical core (TDycore) library

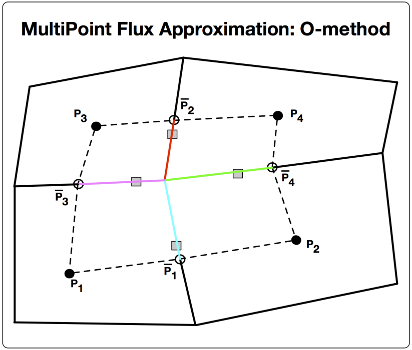
- Developing a scalable **library** on top of PETSc framework
- ELM dycore will be an **application** of the TDycore library
- Open-source and open-development
- Core library is written in C with Fortran bindings
- Supports runtime configurability: **`-tdy_method`** {`wy|mpfao|...`}
- Adopted PFLOTRAN's regression testing framework for the TDycore lib
- Includes 5 demo applications and 27 regression tests
- Available at `https://github.com/TDycores-Project/TDycore`
- Using Travs-CI for regression testing `https://travis-ci.org/TDycores-Project/TDycore`

Terrestrial Dynamical core (TDycore) library

- Developing a scalable **library** on top of PETSc framework
- ELM 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|...}`
- Adopted PFLOTRAN's regression testing framework for the TDycore lib
- Includes 5 demo applications and 27 regression tests
- Available at `https://github.com/TDycores-Project/TDycore`
- Using Travs-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

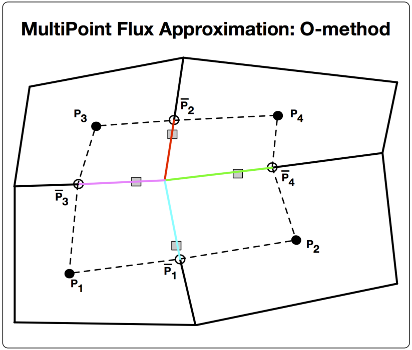
- Used in reservoir simulators with non-orthogonal grids[1]



[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

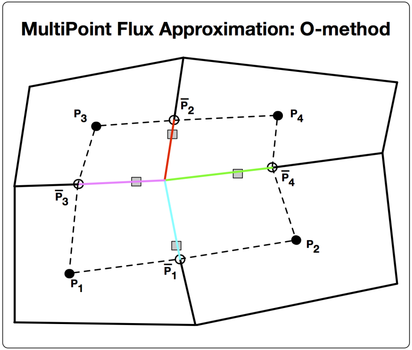
- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method



[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

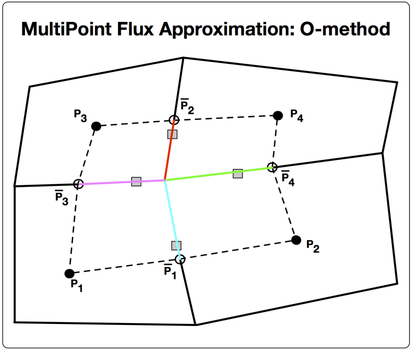
- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method
- Each control volume is subdivided into interactions volumes



[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

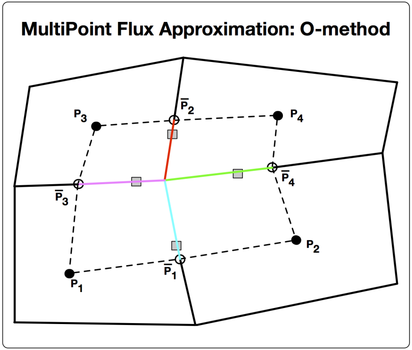
- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method
- Each control volume is subdivided into interactions volumes
- Pressure continuity is assumed at certain locations along the edges



[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method
- Each control volume is subdivided into interactions volumes
- Pressure continuity is assumed at certain locations along the edges

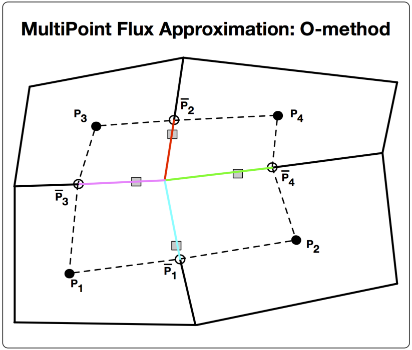


- Flux continuity is enforced along the edges

[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method
- Each control volume is subdivided into interactions volumes
- Pressure continuity is assumed at certain locations along the edges

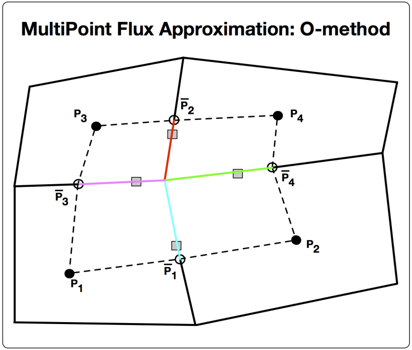


- Flux continuity is enforced along the edges
- Pressure values at the cell center are the unknowns

[1] Aavatsmakr, 2002, Comp. Geosci.

Multi Point Flux Approximation: O-Method

- Used in reservoir simulators with non-orthogonal grids[1]
- Based on finite volume discretization method
- Each control volume is subdivided into interactions volumes
- Pressure continuity is assumed at certain locations along the edges



- Flux continuity is enforced along the edges
- Pressure values at the cell center are the unknowns
- A serial implementation of the method for 2D and 3D grids has been completed

[1] Aavatsmakr, 2002, Comp. Geosci.

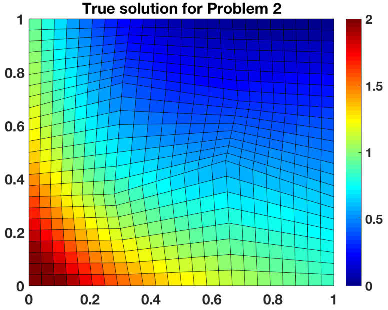
MPFA-O: Results

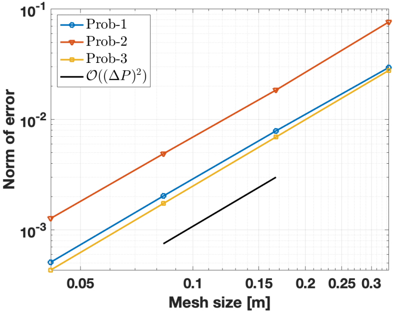
Method of Manufactured Solutions (MMS) is used to verify the implementation for a range of problems

1. $P = 3.14 + x(1-x) + y(1-y)$ and $K = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix}$

2. $P = (x-1)^4 + (1-x)(1-y)^3 + \sin(1-y)\cos(1-x)$ and $K = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix}$

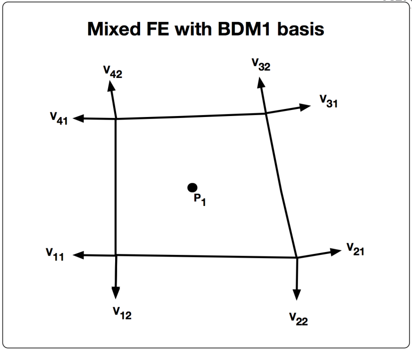
3. $P = x(1-x) + y(1-y) + z(1-z)$ and $K = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \end{bmatrix}$





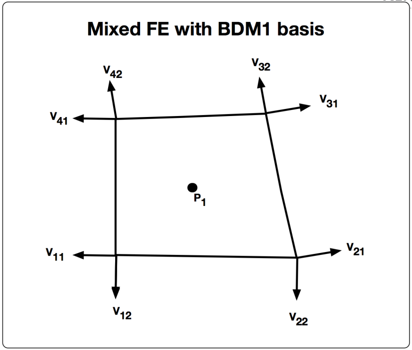
Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids



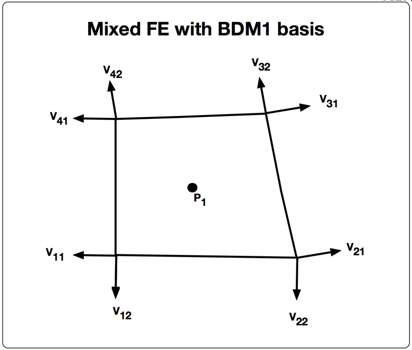
Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids
- Implement the lowest order Brezzi–Douglas–Marini (BDM) space for velocity and constant pressures



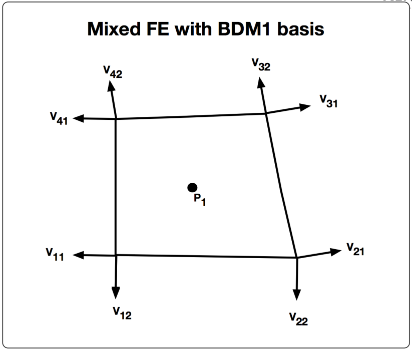
Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids
- Implement the lowest order Brezzi–Douglas–Marini (BDM) space for velocity and constant pressures
- Leads to a saddle point problem, requires specialized preconditioning techniques



Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids
- Implement the lowest order Brezzi–Douglas–Marini (BDM) space for velocity and constant pressures
- Leads to a saddle point problem, requires specialized preconditioning techniques

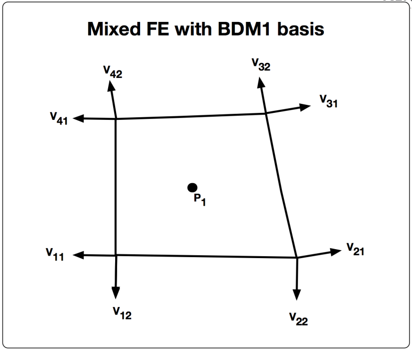


- Also implemented WY method[2], which uses special quadrature to allow for local velocity elimination

[2] Wheeler & Yotov, 2006, SIAM J. Numer. Anal.

Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids
- Implement the lowest order Brezzi–Douglas–Marini (BDM) space for velocity and constant pressures
- Leads to a saddle point problem, requires specialized preconditioning techniques

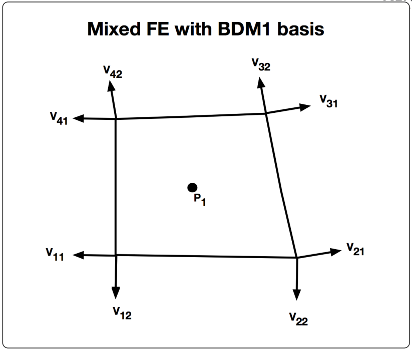


- Also implemented WY method[2], which uses special quadrature to allow for local velocity elimination
- WY method leads to a symmetric and positive definite cell-centered system for the pressures

[2] Wheeler & Yotov, 2006, SIAM J. Numer. Anal.

Mixed Finite Elements

- Handles discontinuous coefficients on non-orthogonal grids
- Implement the lowest order Brezzi–Douglas–Marini (BDM) space for velocity and constant pressures
- Leads to a saddle point problem, requires specialized preconditioning techniques



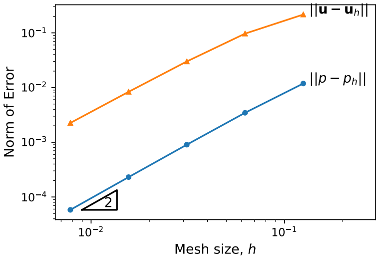
- Also implemented WY method[2], which uses special quadrature to allow for local velocity elimination
- WY method leads to a symmetric and positive definite cell-centered system for the pressures
- Implementation works on distorted grids in 2D/3D in parallel

[2] Wheeler & Yotov, 2006, SIAM J. Numer. Anal.

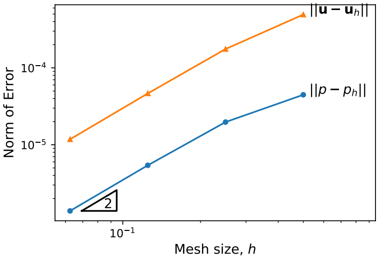
Mixed Finite Elements: Results

- We verify the BDM and WY implementation with the same problems as MPFA-O
- Obtain 2nd order convergence in pressure and velocity on distorted grids in 2D and 3D

Problem 2



Problem 3



Time Integration

- Using PETSc **T**ime-**S**tepping (TS) to avoid hard-coding a time integration schemes

Time Integration

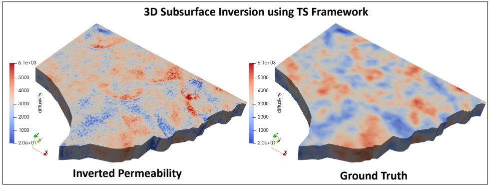
- Using PETSc **T**ime-**S**tepping (TS) to avoid hard-coding a time integration schemes
- TS provides runtime configurability: **`-ts_type`** `beuler|cn|rk|...`

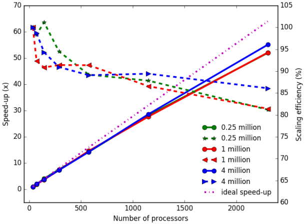
Time Integration

- Using PETSc **T**ime-**S**tepping (TS) to avoid hard-coding a time integration schemes
- TS provides runtime configurability: **`-ts_type`** `beuler|cn|rk|...`
- TS also provides a framework based on discrete adjoints models for **sensitivity analysis**

Time Integration

- Using PETSc **T**ime-**S**tepping (TS) to avoid hard-coding a time integration schemes
- TS provides runtime configurability: **`-ts_type`** `beuler|cn|rk|...`
- TS also provides a framework based on discrete adjoints models for **sensitivity analysis**



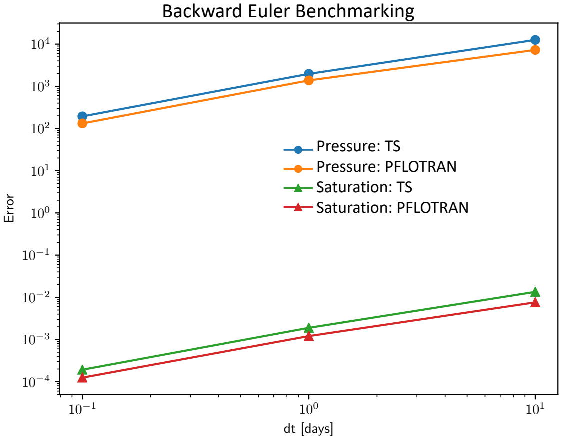


Time Integration: Results

- Implemented PETSc TS to solve non-linear mass and energy balance PDEs in PFLOTRAN

Time Integration: Results

- Implemented PETSc TS to solve non-linear mass and energy balance PDEs in PFLOTRAN
- Error scaling characteristics are similar for the hard-coded PFLOTRAN's backward euler scheme and TS scheme.

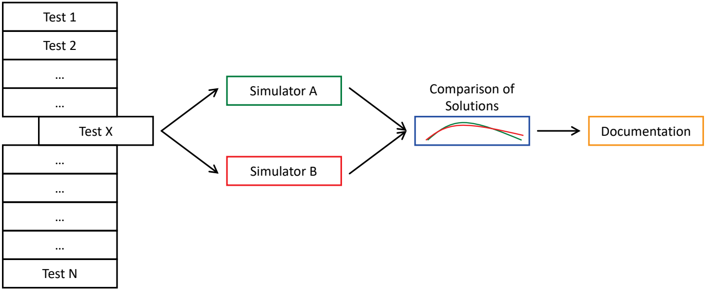


Verification & Validation

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

Verification & Validation

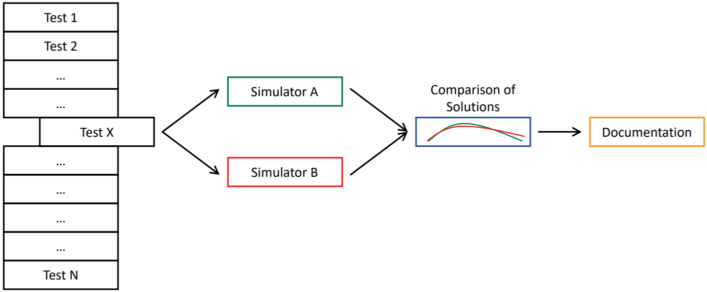
Developing an automated, python-based framework for V&V testing in the cloud that is code-agnostic



- The object-oriented framework compiles a list of tests to be run by a subset of simulators

Verification & Validation

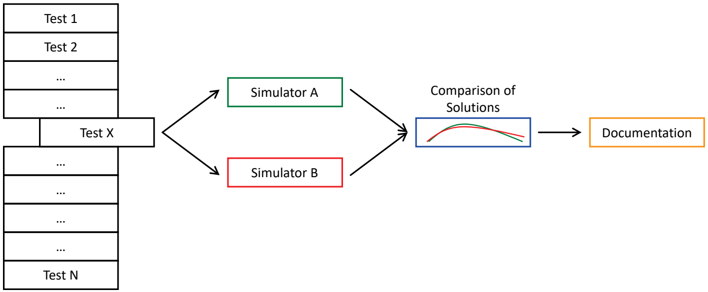
Developing an automated, python-based framework for V&V testing in the cloud that is code-agnostic



- 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

Verification & Validation

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 Sphinx

Benefits of the V&V Framework

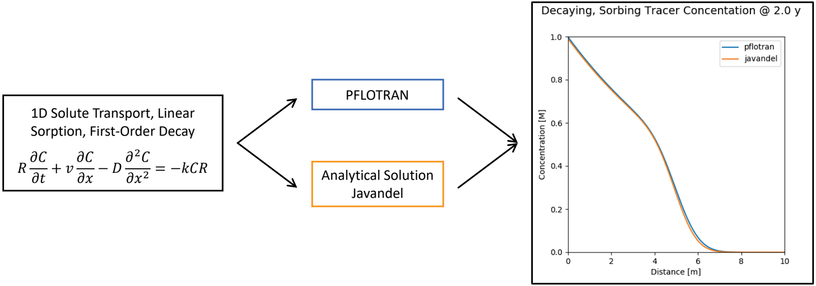
- Confidence: Quality assurance
- Automation: Push-button testing in Cloud
- Maintainability: Python OO design maximizes code reuse and eases future refactoring.
- Longevity: Adoption by other simulation frameworks will better ensure vitality.
  - PFLOTRAN will leverage the same framework.

Example V&V Simulation

Pacific Northwest NATIONAL LABORATORY

1D Solute Transport with Linear Sorption and First-Order Decay

- A recursive search finds a configuration file (.cfg) that specifies that the decay_and_sorption test be run by PFLOTRAN and the Javandel analytical solution.
- An options file (.opt) sets runtime and output options.
- Results are post-processed and plotted with Matplotlib.



Next steps

- Extend the serial implementation of MPFA-O method to support multiple processors

Next steps

- Extend the serial implementation of MPFA-O method to support multiple processors
- Perform an inter-comparison of spatial discretization methods

Next steps

- Extend the serial implementation of 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

Next steps

- Extend the serial implementation of 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 against other models (e.g. PFLOTRAN)

Next steps

- Extend the serial implementation of 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 against other models (e.g. PFLOTRAN)
- Couple the dycore with E3SM Land Model for a watershed scale simulation

Acknowledgements

Funding for this research is provided by the U.S. Department of Energy's Scientific Discovery through Advanced Computing program