PFLOTRAN Development

U.S. Nuclear Waste Technical Review Board
Fall Workshop
November 3-4, 2021

Michael Nole
Sandia National Laboratories

SAND2021-13690 PE
Outline

- **Introduction**: what is PFLOTRAN, and where is it used?
- **Open Source**: software development and computational framework
  - Version Control
  - Task Management
  - Verification Testing
- **Process Modeling**
  - Where PFLOTRAN fits into Geologic Disposal Safety Assessment (GDSA) Framework
  - Process model coupling
  - Advancements over the original code
Introduction: What is PFLOTRAN?

- Scalable, finite volume reactive multiphase flow and transport code for simulating subsurface processes
- Open source license (GNU LGPL 2.0)
- Object-oriented Fortran 2003/2008
  - Pointers to procedures
  - Classes (extendable derived types with member procedures)
- Founded upon well-supported open source libraries
  - MPI, PETSc, HDF5, METIS/ParMETIS/CMAKE
- Demonstrated performance
  - Maximum # processes: 262,144 (Jaguar supercomputer)
  - Maximum problem size: 3.34 billion degrees of freedom
  - Scales well to over 10K cores
Introduction: Where is PFLOTRAN used?

- **Nuclear waste disposal**
  - Waste Isolation Pilot Plant (WIPP) in Carlsbad, NM: *underwent rigorous Quality Assurance for qualification as an official WIPP PA flow code (July 2021)*
  - US DOE NE Spent Fuel and Waste Science and Technology (SFWST)
  - DEvelopment of COupled models and VALidation against EXperiments (DECOVALEX): international model comparison collaboration
  - Forsmark Spent Fuel Nuclear Waste Repository (Sweden, Amphos21)

- **Climate: coupled overland/groundwater flow**
  - Next Generation Ecosystem Experiments Arctic
  - DOE Earth System Modeling Program

- **Biogeochemical transport modeling**
  - U transport at Hanford 300 Area
  - Hyporheic zone biogeochemical cycling
    - Columbia River, WA, USA
    - East River, CO, USA
Open Source Framework

- **Benefits**
  - Collaboration: development, testing, and debugging
  - Transparency: exposes implementation details critical to scientific reproducibility, but excluded by journal publications
  - Lower barrier to entry (none if you have the expertise)
  - Code fitness must be maintained to survive
Open Source Framework

- Public code repository: [https://bitbucket.org/pflotran/](https://bitbucket.org/pflotran/)
  - Version control
  - Development philosophy and coding standards
  - Merge request requirements and mandatory checks
  - Major/minor/patch versioning
- Documentation: [https://www.pflotran.org/documentation/](https://www.pflotran.org/documentation/)
- Continuous integration
  - Regression testing
  - Unit testing
- Task Management
  - Jira
- QA Test Suite: [https://www.pflotran.org/qa/](https://www.pflotran.org/qa/)
  - Modular design
Open Source Framework

- **Version Control**

  ![Diagram showing Bitbucket interface]

  - **Author**
  - **Unique code change identifier**
  - **Code change description**
  - **Verification of successful unit/regression testing**

<table>
<thead>
<tr>
<th>Author</th>
<th>Commit</th>
<th>Message</th>
<th>Date</th>
<th>Builds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jenn Frederick</td>
<td>2fb9616</td>
<td>jenn/nwt-debugging-start-fresh</td>
<td>20 hours ago</td>
<td>✔</td>
</tr>
<tr>
<td>Jenn Frederick</td>
<td>3189240</td>
<td>jenn/nwt-debugging-start-fresh</td>
<td>20 hours ago</td>
<td>✔</td>
</tr>
<tr>
<td>Glenn Hammond</td>
<td>d2a1555</td>
<td>Added hea... glenn/inversion-cgl</td>
<td>20 hours ago</td>
<td>✔</td>
</tr>
</tbody>
</table>
Open Source Framework

- Task Management
  - Developer assignment
  - Development stage
  - Prioritization
  - Issue type
  - Relative effort
  - Scope re-evaluated bi-weekly
Open Source Framework

- Code Verification Testing: GDSA Quality Assurance (QA) Test Suite
  - Modular and extendable
  - Tests against analytical solutions and outputs from GDSA-QA Test Suite

Main Folder:
- Makefile
- Makefile to run QA tests using QA Toolbox
- <test>.cfg
- Text file with paths to <test>.cfg
- <simulators> sim
- Text file with paths to simulator executed
- tests
- Docs
- HTML

QA-Toolbox
- Python Harness
  - Simulator1
  - Simulator2
  - ...
  - SimulatorN

Results from QA-Toolbox get outputted as reStructuredDoc

Gitlab
Process Modeling: GDSA Framework

Next Gen Workflow

- Input Parameters
  - Parameter database

- Uncertainty Sampling and Sensitivity Analysis

- Computational Support
  - Processing
    - VorCrust
    - dfnworks
  - Visualization
    - ParaView

Multi-Physics Simulation and Integration

- Source Term and EBS Evolution Model
  - Inventory
  - Decay, ingrowth
  - WF degradation
  - WP degradation
  - Radionuclide release
  - Thermal, mechanical
  - Gas generation

- Flow and Transport Model
  - Advection, diffusion, dispersion
  - Discrete fracture networks
  - Multiphase flow
  - Sorption, solubility, colloids
  - Isotope partitioning
  - Decay, ingrowth
  - Thermal effects
  - Chemical reactions

- Biosphere Model
  - Exposure pathways
  - Uptake/transfer
  - Dose calculations

Results
Process Modeling: GDSA Framework

- **Fluid “flow” modes:**
  - RICHARDS: conservation of water mass, variably saturated flow
  - TH: thermo-hydro; conservation of water mass and conservation of energy
  - GENERAL: conservation of water and air mass and conservation of energy; miscible multiphase flow

- **Solute “transport” modes:**
  - GIRT: global implicit reactive transport
  - UFD Decay: radionuclide sorption, partitioning, decay, and ingrowth
  - NWT: nuclear waste transport; different primary independent variables from GIRT or UFD Decay
Process Modeling: Process Model Coupling

- **Traditional Time-stepping Loop**
  - Initialization
  - Time Stepping
  - Flow
  - Reactive Transport
  - Done?
  - Finalization

- **PFLOTRAN Workflow**
  - Initialization
  - Execution
  - Finalization
  - Process Model Couplers
Process Modeling: Process Model Coupling

Process Model Coupler

- Process Model
  - Multiphase Flow

- Numerical Methods
  - Time Integrator
  - Newton Solver
  - Linear Solver

Peer (sync-point)

Child (catch-up)
PMC = Process Model Coupler

PMC A

PMC B

PMC C
Process Modeling: Process Model Coupling

PMC = Process Model Coupler

PMC A → PMC B → PMC M → PMC Y

PMC C → PMC M → PMC Z
Process Modeling: Process Model Coupling

Radioactive Waste Process Model Coupling

Initialization

Execution

Finalization

Multiphase Flow

Transport

Waste Form/Package

Isotope SPDI
Benefits

- Customizable linkage between process models, e.g.
  - Flow
  - Transport
  - Reaction
  - Updates to material properties at select times
- Flexible time stepping
  - Individual processes may run at their own time scale.
- Modularity for incorporating new process models
  - Time stepping loops for existing process models are not impacted.
Process Modeling Advancements

- Multiphase fluid and heat flow
- Radioactive sorption/partitioning/decay/ingrowth model (UFD Decay)
- Soil matrix compressibility
- Flexible models for thermal conductivity and anisotropy
- Improved multiphase capabilities during dry-out

Price et al., 2021

Nole et al., 2021
Process Modeling Advancements

- Sorption isotherm generalization
- Fuel Matrix Degradation Model (FMDM)
- Biosphere well model
- Multi-continuum transport
- Advanced linear and nonlinear solvers
- High temperature equations of state
- Reduced order geomechanics models

LaForce et al., 2021  
Chang et al., 2021
References