IN11C-1471

Loosely coupled hybrid subsurface simulations using the Swift scripting system

Karen Schuchardt, Tim Scheibe, Jared Chase, Khushbu Agarwal, Bruce Palmer, Alexandre Tartakovsky, George Redden

karen.schuchardt@pnnl.gov

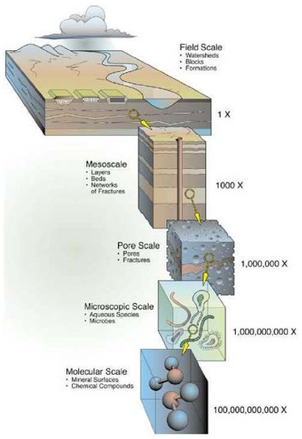
Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

Overview

Recent studies have pointed to the need for pore-scale resolution (mm scale) of reactive transport processes for accurate simulation of some types of problems. Of particular note are processes which are controlled by mixing (primarily diffusion) at interfaces between fluid and/or solid phases containing the various reactants. Other problems will require applications at the microscopic and even molecular scales. Since it is not computationally feasible to simulate even moderate sized domains at these scales, we are developing coupled hybrid simulators which model the full domain at coarse scales and selected subdomains at finer scales. Our first hybrid simulator applies a continuum model over the full domain and a pore scale model in selected regions of high reactivity.

Our hybrid simulator involves executing pore scale simulations at each iteration of the continuum model [Scheibe 2008]. Each pore scale iteration involves multiple separate pore scale domains, each executing as an MPI application thus involving multi-level parallelism. Our simulator is coupled via the Swift scripting system [Wilde 2009] and executes in a High Performance Computing (HPC) environment. We describe the implementation of the hybrid simulator, initial use cases, and results of our demonstration.

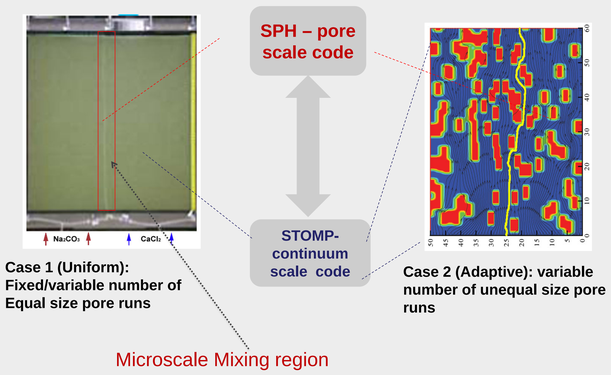


Use Cases

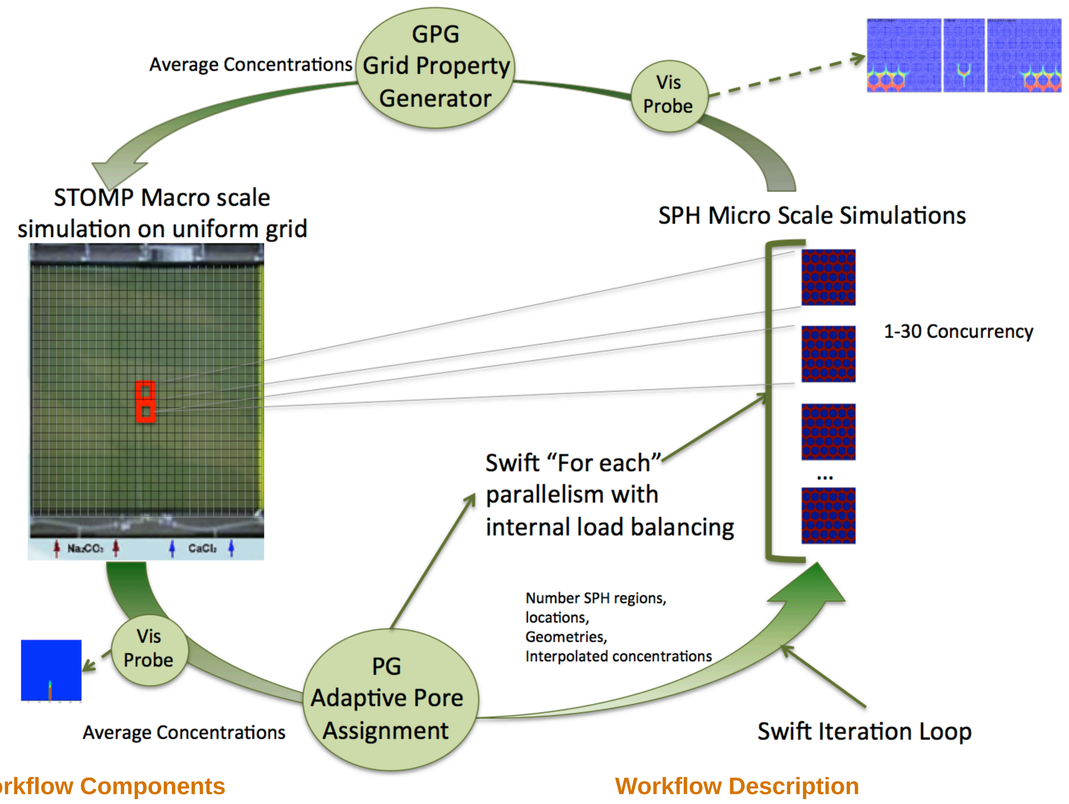
Our demonstration problem will mimic meso-scale experiments conducted by George Redden (INL) involving a mixing reaction in a quasi-3D cell. Non-specific reactants A and B are injected on the left and right side of the domain respectively and mix to form C along the center line. Initially, we restrict our simulation to the uniform use case (Use Case 1 below).

Use Case 1 (Uniform): The system is filled with uniform sand. Thus, the mixing region is known and can be divided into equal size pore domains that match continuum grid cells. Initially, the reaction starts at the bottom of the domain where the injections occur. As the reaction proceeds, C will form and move vertically up the domain. We can determine when new SPH regions should be initiated using a threshold of the value of C. Similarly, as the reaction completes, SPH regions can be eliminated from the bottom. The pore simulation zone will be like a moving window that moves up the system

Use Case 2 (Adaptive): The experimental system is packed heterogeneously with two different types of sand having different hydraulic conductivities and porosities (red is low conductivity and blue is high), thus resulting in non-uniform unequal sized pore domains. The yellow line indicates a likely reaction path. This use case creates many more challenges for the both the mathematical integration of the scales and for load balancing due to pore region size and location variability.

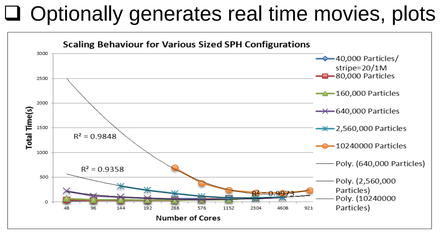


Hybrid Subsurface Model



Workflow Components

- A Swift workflow is used to couple our workflow consisting of
  - Subsurface Transport Over Multiple Phases (STOMP [White 2010])
    - Serial – models full domain
  - Smoothed Particle Hydrodynamics (SPH) [Palmer 2007] applied to localized mixing regions.
    - Parallel – multiple pore domains simulated concurrently
  - The Pore Generator(PG) (python)
    - Determines location/number of pore regions to run
    - Prepares SPH inputs, using STOMP data and the previous SPH run data if exists
  - Grid Parameter Generator(GPG) (python)
    - Computes average concentrations and concentration rates for next STOMP iteration
  - Visit
    - Optionally generates real time movies, plots



Workflow Description

- Defines wrappers to run each of the component applications
- Runs a predefined number of loops over the entire workflow (determined empirically)
- Sets up file dependencies between components so that Swift can automatically schedule tasks. Many of the file dependencies are dynamically determined
- Uses Swift foreach construct to parallelize SPH runs within an iteration
- The number of pore scale runs may change at every iteration and may represent pores of equal (Use Case 1) or unequal size (Use Case 2).

Special Swift Customizations

- **Load Balancing** : Determines optimal minimum (minprocs) and maximum(maxprocs) processors count to run SPH, based on domain size based on SPH scaling studies (image to left).
- **Adaptive Scheduling :** Manages multiple batches of SPH runs when insufficient processors are available using 'minprocs' per SPH; otherwise divides processes equally amongst SPH runs, while enforcing 'maxprocs' limit.
- Uses aprun to initiate all components

Experiment

Configuration

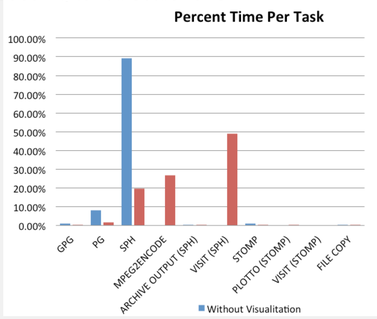
- 2D system 30 cm x 31 cm
- STOMP .5cm grid spacing
- .5 cm x .5cm uniform SPH sand grains (.7mm) resulting in 40,000 particles per pore region
- Time step of 10 seconds in STOMP and SPH

Execution Environment

- Hopper batch jobs using PBS
- Swift 0.94 RC (r6065)
- Serial STOMP due to small initial problem size
- Min 48 processors/SPH task
- Max 96 processors/SPH task
- 500 Stomp iterations
- New SPH determined by (conc_A *conc_B)/(MAXCONC_A *MAXCONC_B) > tolerance
- Drop bottom SPH (that reach steady state) when more than 4 become active.

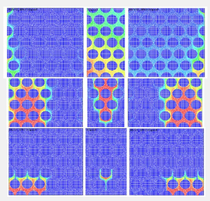
Result Summary

- Run time: 10 hours
- 300+ STOMP iterations
- 1200+ SPH simulations
- ~ 2GB/s average bandwidth (H5Part)
- Analyzing data and coupling algorithm before proceeding to full scale runs

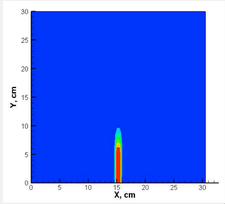


Summary and Future Work

- The loosely coupled approach to hybrid model mitigates the complexities that arise from complete integration of two highly complex numerical codes. It also enables us to switch codes and coupling mechanisms during this highly exploratory phase of hybrid model development.
- Defining and testing the coupling mechanism, posed a significant challenge. This resulted in multiple versions of the coupling routines with test and evaluation cycles.
- Some of the mechanisms we developed are more applicable to large problems and use case 2.
- Swift is portable and will work on many platforms so we can readily transport this to other environments. Challenges we faced:
  - Thinking in terms of file dependencies instead of procedural code challenging initially
  - Debugging was difficult initially. We worked with Swift team to improve diagnostics
  - More error reporting is needed – for example references to file name strings do not result in expected behavior and script writer should be warned.
  - Restart function is desired.
- Performance of the workflow over a large run is shown at the left. Red bars are results when visualization processing iwas done in realtime, whereas blue bars are from a run where data was post processed. In future, we plan to explore ways to mitigate the cost of inline analysis.
- Overhead due to the scripting framework was negligible
- I/O is not an issue for the initial problem. However, will need to be addressed for larger (eg 3D) simulations. Our python scripts incur costs using h5dump command and could be converted to direct HDF5 library calls.



Visualization of 3 stacked pore scale regions (left) and continuum domain (right) showing the reaction progression.



References

1.Scheibe, T. D., A. M. Tartakovsky, D. M. Tartakovsky, G. D. Redden, and P. Meakin, 2007. Hybrid numerical methods for multiscale simulations of subsurface biogeochemical processes, *Journal of Physics: Conference Series* 78: 012063.

2.Wilde, M., Foster, I., Iskra, K., Beckman, P. Zhang, Z. Espinosa, A., Hategan, M., Clifford, B., Raicu, L., "Paralel Scripting for Applciatinos at the Petascale and Beyond", Computer, Vol. 42, No 11, 2009.

3.White, M.D., and M. Oostrom. 2006. STOMP Subsurface Transport Over Multiple Phases, Version 4: User's Guide PNNL-15782 (UC-2010).

4.Palmer, B. J., V. Gurumoorthi, A. M. Tartakovsky, and T. D. Scheibe, "A Component Based Framework for Smoothed Particle Hydrodynamics Simulations of Reactive Fluid Flow in Porous Media," *International Journal of High Performance Computing Applications* 24(2):228-239, 2010.

PNNL-SA-92340