Procedia Computer Science

Volume 51, 2015, Pages 1098–1107

ICCS 2015 International Conference On Computational Science

A Hybrid Multiscale Framework for Subsurface Flow and Transport Simulations

Timothy D. Scheibe[1*], Xiaofan Yang[1], Xingyuan Chen[1], Glenn Hammond[2]

[1]*Pacific Northwest National Laboratory, PO Box 999, MS K9-36, Richland, WA 99352*

[2]*Sandia National Laboratory, Albuquerque, NM*

tim.scheibe@pnnl.gov

Abstract

Extensive research is aimed at improving predictive ability of biogeochemical earth and environmental system simulators, with applications ranging from contaminant transport and remediation to impacts of carbon and nitrogen cycling on local ecosystems and climate. Most process-based numerical models are designed for a single characteristic length and time scale. For application-relevant scales, it is necessary to introduce approximations and empirical parameterizations to describe complex systems because of limitations on process understanding, system characterization and computation. Using emerging understanding of biological and environmental processes at fundamental scales to advance predictions of the larger system behavior requires the development of multiscale simulators, and there is strong interest in coupling microscale and macroscale models together in a hybrid multiscale simulation. A limited number of hybrid multiscale simulations have been developed for biogeochemical systems, mostly using application-specific approaches for model coupling. We are developing a generalized approach to hierarchical model coupling designed for high-performance computational systems, based on the Swift computing workflow framework. In this presentation we will describe the generalized approach and provide two use cases: 1) simulation of a mixing-controlled biogeochemical reaction coupling pore- and continuum-scale models, and 2) simulation of biogeochemical impacts of groundwater – river water interactions coupling fine- and coarse-grid model representations. This generalized framework can be customized for use with any pair of linked models (microscale and macroscale) with minimal intrusiveness to the at-scale simulators. It combines a set of python scripts with the Swift workflow environment to execute a complex multiscale simulation utilizing an approach similar to the well-known Heterogeneous Multiscale Method. User customization is facilitated through user-provided input and output file templates and processing function scripts, and execution within a high-performance computing environment is handled by Swift, such that minimal to no user modification of at-scale codes is required.

Keywords: hybrid multiscale simulation; pore scale; biogeochemistry; reactive transport; high-performance computing

* Corresponding author, tim.scheibe@pnnl.gov, +001-509-371-7633

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.276

1 Introduction

One of the most significant challenges in earth systems modeling is the large disparity between the spatial and temporal scales at which fundamental flow, transport, and reaction processes can best be understood and quantified (e.g., microscopic to pore scales and seconds to days) and those at which practical model predictions are needed (e.g., aquifer to watershed scales and years to centuries). While the multiscale nature of earth systems is widely recognized, technological limitations in computation and characterization restrict most practical modeling efforts to fairly coarse representations of heterogeneous properties and processes. For some modern problems, the necessary level of simplification is such that model parameters may lose physical meaning and model predictive ability is questionable for any conditions other than those to which the model was calibrated. Recently, there has been broad interest across a wide range of scientific and engineering disciplines in simulation approaches that more rigorously account for the multiscale nature of systems of interest. In the past decade, methods that connect multiple models defined at distinct scales (typically with different mathematical representations of physical, biological and chemical processes) have begun to be applied. We refer to these approaches as hybrid multiscale methods; Scheibe et al. (2015) provide a review of several classes of multiscale methods including hybrid multiscale methods, with discussion of recent applications to subsurface flow and reactive transport simulation. These applications have used customized approaches to model coupling, specifically tailored to the macroscale and microscale simulators and problem of interest.

Recently there has been increased interest in developing more general and extensible frameworks for multiscale model coupling (e.g., Falcone et al. 2012; Borgdorff et al. 2013; Tang et al. 2014); see reviews by Yang (2013) and Groen et al. (2013). We have developed a parallel workflow structure for multiscale modeling based on the Swift workflow environment (Katz et al. 2011) that manages the complex process of executing many coupled microscsale and macroscale code runs in a parallel computing environment over the course of a single integrated multiscale hybrid simulation. We initially applied this approach to a specific problem using problem-specific scripts for data exchange and model coupling (Scheibe et al. 2014). We are generalizing this approach to allow loose coupling of any two codes (macroscale and microscale) using a hierarchical multiscale approach. The generalization employs user-specification of input and output file templates from each simulator, together with functional descriptions to be applied to transform output data from one scale into input files for the other scale, in a Swift-controlled scripting environment. Here we describe the generalized multiscale modeling framework which is currently being developed and tested using two realistic use cases: 1) a mixing-controlled reaction similar to that simulated in Scheibe et al. (2014), in which pore-scale and continuum-scale models are loosely coupled; and 2) a multiscale simulation of groundwater – river water mixing and its impacts on biogeochemical cycling of carbon and nitrogen in the hyporheic zone of a large river.

2 Hybrid Multiscale Simulation

2.1 General Approach

We consider “microscale” simulation of flow, reaction, and transport at which fundamental processes are more accurately represented, and “macroscale” simulation at which continuum processes are represented in an averaged sense. Macroscale process descriptions and parameters can be defined by directly averaging microscale processes and parameters, but in general this does not provide a benefit as complete microscale information is required to perform the averaging. Upscaling methods introduce a scaling law (Wood 2009) that allows a closure approximation to be made in which macroscale processes and parameters can be posed in terms that do not require explicit pore-scale

information. For example, under certain conditions macroscale dispersion is a function only of porosity and does not depend explicitly on pore-scale flow geometry. However, we know that there exist some conditions (posed as a function of non-dimensional Damkohler and Peclet numbers) under which the assumptions required for a general closure are violated (Battiato et al. 2009; Battiato and Tartakovsky 2011; Boso and Battiato 2013). Since it is computationally too expensive to model microscale processes over a large domain, and the proportion of the domain over which continuum approximations are violated is relatively small, solving the microscale model only within the sub-domain and using the continuum model over the remainder of the domain offers potential to obtain a reasonable balance between computational requirements and solution accuracy. Many hybrid multiscale methods apply the microscale model over a sub-domain for the complete simulation time, and couple it with the macroscale model at boundaries or in an overlapping domain. However, a hierarchical dimension reduction approach (Tartakovsky and Scheibe, 2011) offers the potential for further computational efficiency and eliminates the need for boundary condition matching (which is especially challenging in the case where the microscale and macroscale models have significantly different formulations, e.g., particle-based lagrangian vs. grid-based eulerian).

Figure 1 presents the general approach and workflow of the hybrid multiscale method that provides a loose coupling between macroscale and microscale simulations. The approach is based on the Heterogeneous Multiscale Method (HMM) for hierarchical multiscale simulation (E et al. 2003) as implemented in the Dimension Reduction with Numerical Closure method of Tartakovsky and Scheibe (2011). In Figure 1, grey boxes represent the simulation codes to be coupled (two different scales, microscale and macroscale). Green ovals represent python scripts that perform data transfer between the two scales (i.e., perform the Restriction and Lifting operators of the HMM). Yellow boxes represent user-provided templates that describe the format of input and output files from both simulators as well as adaptivity rules and functions for processing output data from one simulation scale to generate input data for the other simulation scale. Blue boxes represent the script-generated input and output files, created from the user-provided templates by the Restriction Operator (RO) and Lifting Operator (LO) scripts. Optionally, adaptivity rules to determine when and where microscale simulations are active within the macroscale domain can be specified by the user and are implemented by the Adaptivity Manager (AM) script. The functionality for interpreting input and output file templates and applying processing functions are provided by FORTRAN-90 modules from the JUPITER API (Banta et al. 2006). The JUPITER API provides comprehensive methods for interacting with multiple process models, requiring only that they have text-only input and output files and can be invoked by an operating-system command, designed for improved communication between multiple applications. It consists of eleven modules that provide encapsulation of data and operations on that data, and has been used to construct a number of earth science applications including the Universal Inverse Code UCODE (Poeter et al. 2005).

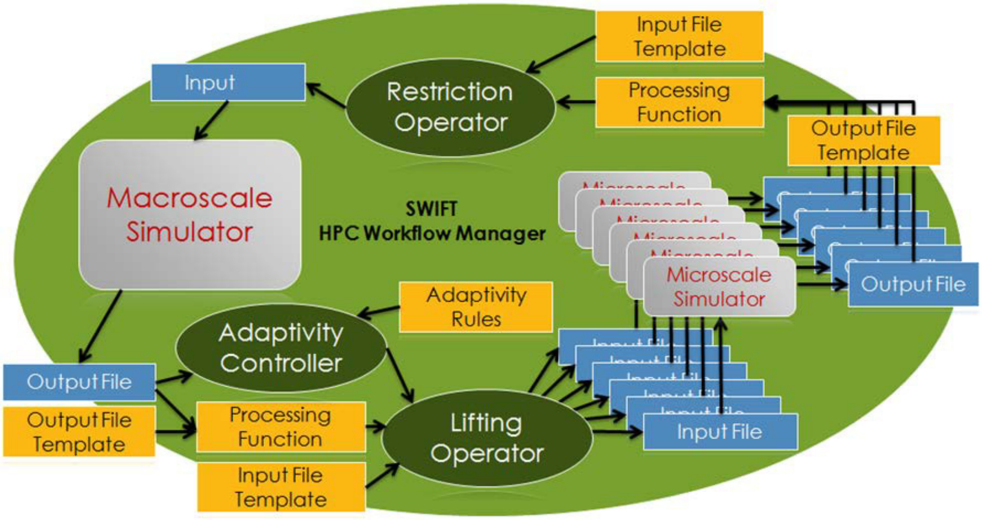


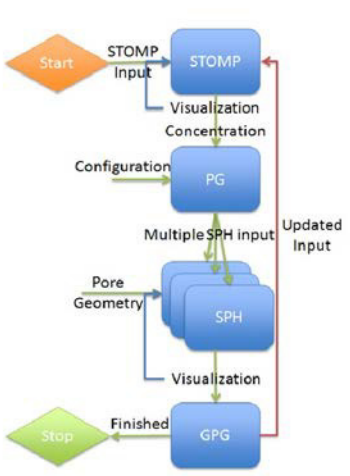
Figure 1 Schematic diagram of the generalized hybrid multiscale simulation framework.

2.2 Swift-Controlled Workflow

The hybrid model workflow is executed and managed using the Swift workflow environment (Wilde et al. 2011). It is launched by invoking a single instance of Swift, along with a request to allocate all the resources (computational nodes) anticipated to be needed during the entire course of the execution, which eliminates the need for each independent macroscale and microscale simulation to wait in the supercomputer queuing system multiple times. The workflow is adaptive and portable. It supports dynamic scheduling of tasks, and utilizes Swift's logging and error handling capabilities. Post-processing functions are also supported by the workflow including visualization and provenance tracking capabilities.

The hybrid task workflow is presented in Figure 2a, which follows the conceptual model explained in Sec. 2.1 and consists of four main modules: 1) a parallel (or serial, depending on computational demands) macroscale simulator covering the full computational domain, 2) a serial python script – Lifting Operator (LO) – that adaptively determines microscale simulation regions, executes the reconstruction step, and constructs microscale simulator input files, 3) multiple instances of a parallel microscale simulator, and 4) a serial python script – Restriction Operator (RO) – that performs the numerical closure and constructs macroscale input files. The macroscale simulator is used to advance the macroscale process simulation for a specified period of time. Configuration files describing the initial model configuration are provided to initiate the simulation. The LO script provides algorithms to reconstruct initial conditions for microscale subdomains based on macroscale quantities from the previous macroscale simulator time step. It is also potentially linked with an Adaptivity Manager that determines how many and which microscale subdomains require simulation based on user-specified adaptivity rules. The microscale simulator is then executed to perform microscale simulations for each active subdomain. The RO script then creates an updated macroscale simulator input file based on output from microscale runs and the process iterates.

Our Swift workflow consists of an application for of the modules. A foreach construct is used to run all microscale simulations in parallel, as per our adaptive scheduling policy. A hybrid_model function consists of all application components, and defines a single iteration of the workflow. An iterative loop over the hybrid_model function is used, enforcing serial execution between iterations, where outputs from one iteration serve as input to the next iteration. A maximum number of iterations is specified at the command line by the user. Swift is configured to run locally on the system and definitions are provided (path to code executables) for each of the applications in the workflow. A sample of the Swift workflow code executed in Use Case 1 is shown in Figure 2b.




```
// ------------------------------  The APPs  ------------------------------
// APP for Stomp
(file plot_output,file output,file stomp_plot, file restart_output) runStomp
    (file input,file zon,file forward,file restart, file conca_dat, file concb_dat,
    file concc_dat, string path) {
   app {stomp @filename(input) @filename(zon) @filename(forward) @filename(restart) @filename(conca_dat)
       @filename(concb_dat) @filename(concc_dat) path stdout=@output stderr=@output;}
   }
}
// App for first PG run which just determines how many SPH to start
(file num, file activesph,file pg_stdout) count_sph(file plot_input, file restart, float tol, int iter,
    file mdcfg, int prev_num_sph) {
   app {pg "-p" @filename(plot_input) "-s" @filename(restart) "-t" tol "-f" 1 "-i" iter "-n"
       prev_num_sph "-m" @filename(mdcfg)  stdout=@pg_stdout stderr=@pg_stdout;
   }
}
// App for PG
(file mdcfg_files[],file grid_val_files[],file src_distr_files[], file pg_stdout) pg(file plot_input,
   file restart, float tol, int iter, file mdcfg, int prev_num_sph, file previous_mdcfg_files[],
   file previous_h5part_files[]) {
   app {pg "-p" @filename(plot_input) "-s" @filename(restart) "-t" tol "-f" 0 "-i" iter "-n" prev_num_sph" -m"
       @filename(mdcfg) "-l" @filenames(previous_mdcfg_files) @filenames(previous_h5part_files)
       stdout=@pg_stdout stderr=@pg_stdout;
   }
}
// App for SPH
(file restart, file tarfile, file movie, file output) runSph (file sph_in, file mdcfg, file dens_profile,
    file grid_val, file src_distr, int procsTask, string path) {
   app {sph @filename(sph_in) @filename(mdcfg) @filename(dens_profile) @filename(grid_val) @filename(src_distr)
       procsTask path stdout=@output stderr=@output;
   }
}
// App for GPG
(file out, file gpg_stdout, file conca_dat, file concb_dat, file concc_dat) gpg (file map, file h5part_files[],
    int runIter, file plotfile) {
   app {gpg "-m" @filename(map) "-r" runIter "-p" @filename(plotfile) "-f" @filenames(h5part_files)
       stdout=@gpg_stdout stderr=@gpg_stdout;
   }
}
```

(b)

Figure 2 (a) Swift-controlled hybrid workflow; (b) a sample of Swift workflow code in Use Case 1

2.3 Use Case 1: Mixing-Controlled Reaction in a Macroscopically Homogeneous System

Our use case 1 (Figure 3) simulates an irreversible mixing-controlled kinetic reaction occurring at the interface between two solutes (reactants), generating a third solute (product). The system is filled with a saturated homogeneous porous medium (sand). Two solutes (denoted as A and B) are injected at the bottom and flow upward to the top at a specified rate. The mixing zone of the two solutes is along the centerline, leading to reaction and formation of the product (C). The rate of reaction at the interface is strongly controlled by the rate of lateral diffusion of the two reactants. The mathematical approach for coupling the pore- and continuum-scale simulations is described in Tartakovsky and Scheibe (2011).

The 2D porous medium system is 30.5 cm x 30 cm, as shown in Figure 3, and is similar to the experiment described in Tartakovsky et al. (2008). The macroscale simulator in use case 1 is a serial version of the Subsurface Transport Over Multiple Phases (STOMP) simulator (Nichols et al. 1997), used to model continuum-scale saturated water flow, solute transport, and reaction in the entire domain. The macroscale STOMP simulations use a regular mesh of size 61 x 60 cells. The microscale simulator uses the particle-based Smoothed Particle Hydrodynamics (SPH) method to solve pore-scale

water flow, solute transport, and reactions in selected sub-domains (Palmer et al. 2010). Each SPH model geometry is homogenous with a size of 0.5 x 0.5 cm (corresponding to a single STOMP grid cell) and containing 40,000 particles. Physical flow and transport processes are represented in a fundamentally different manner in the two model scales. In the macroscale (STOMP) simulations, flow is represented using Darcy's law and conservation of mass, and transport is represented using the advection-dispersion equation. In the microscale (SPH) simulations, Navier-Stokes flow equations are solved explicitly and solute transport is represented as an advection-diffusion process. Because the macroscale domain is relatively small, STOMP executes in serial on a single processor. On the other hand, the SPH code (which is computationally more demanding) executes in parallel on an optimal number of available processors, which is determined during the workflow. A specified flux boundary condition is applied at the bottom of the macroscale domain, with a Darcy velocity of 1 cm/min, and a specified pressure is imposed at the top of the macroscale domain simulating the free outflow boundary of the experiment. No-flow conditions are specified at right and left boundaries.

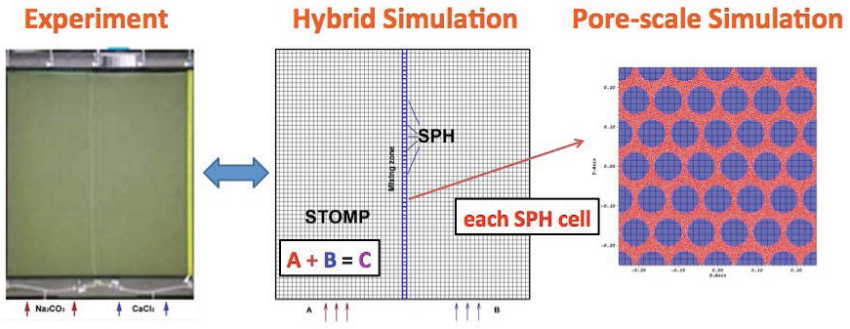


Figure 3 Use case 1: mixing-controlled kinetic reaction in homogeneous porous system

The scripts that adaptively determine subdomain regions (Adaptivity Controller AC) and perform lifting (LO) and restriction (RO) operators are serial and coded in python. The pore-scale (microscale) simulations (SPH) are initiated in a given cell along the central vertical column of macroscopic grid cells (the reaction interface) when sufficient concentrations of A and B exist to initiate reaction, and are turned off once the system locally reaches a quasi-steady state. A complete simulation involves executing many iterations of the hybrid model. In our numerical experiment, the entire simulation consists of ~800 iterations (with no more than 60 pore-scale simulations perfomed during each iteration) and is run on 1536 nodes (24 processors each) on a Cray XE6 system (Hopper, NERSC). It takes ~96 hr as wall clock time to finish the simulation. 85% of run time is used for SPH runs and less than 10% is used for STOMP simulations.

2.4 Use Case 2: Biogeochemical hot spots in the river-groundwater interaction zone

The second use case simulates the effect of interactions between river water and groundwater on biogeochemical reactions in a region of the surficial aquifer adjacent to a large river. It is based on a field research site at the 300 Area of the U. S. Department of Energy's Hanford Site in southeastern Washington state, bordered by the Columbia River (Zachara et al. 2013). The site is located in a semi-arid region, and the subsurface environment is generally low in organic carbon. However, the river is a source of organic carbon, which when mixed with nitrate-rich waters in the aquifer creates the potential for relatively high biogeochemical activity in the sediments immediately adjacent to the river (the hyporheic zone). A region of approximately 400 x 400 meters is of interest because of a)

intrusion of river water during high river stage several hundred meters into the aquifer, and b) the existence of contaminant plumes in the aquifer in this area. However, biogeochemical activity (which can impact the fate of contaminants as they move toward the river) is high in a relatively thin zone (less than 1 meter thick) immediately adjacent to the river (Figure 4). Furthermore, the biogeochemical reactions and flow permeability are strongly impacted by local heterogeneity in the hyporheic zone sediments (referred to as the mud layer), requiring high spatial resolution of processes and material properties within this thin zone. It is not computationally feasible to resolve the entire domain of interest at the same spatial grid resolution needed in the mud layer, and the biogeochemical reaction network needed in the mud layer is much more complex than that needed in the remainder of the large domain.

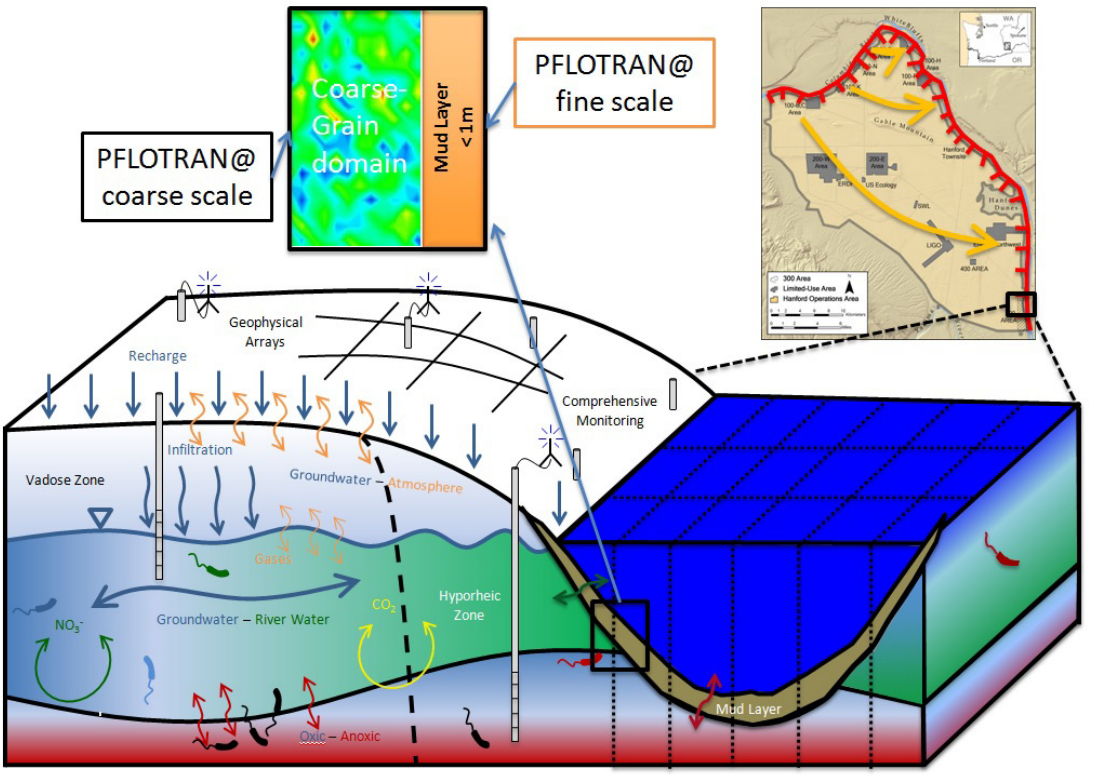


Figure 4 Schematic diagram of use case 2. The upper right diagram shows the location of the 300 Area site in the context of the Hanford Site and the adjacent reach of the Columbia River. The main portion of the figure shows a 3D schematic of the region of interest, intersecting the aquifer and the mud layer adjacent to the river. The upper left diagram shows a local portion of the model domain containing the mud layer and a coarse-grained aquifer domain.

In our hybrid multiscale simulation of this problem, both model scales (microscale and macroscale) are represented using the PFLOTRAN code (Lichtner et al. 2013). However, at the microscale a finely-resolved model grid is employed together with a complex biogeochemical reaction network, while at the macroscale a simplified reaction network is utilized with a coarsely-resolved model grid. The hierarchical approach shown in Figure 1 is used to couple multiple parallel instances of PFLOTRAN defined on microscale sub-domains with a single parallel macroscale instance of PFLOTRAN defined over the full domain.

3 Model Evaluation

Use case 1 has previously been implemented using data transfer and model coupling scripts custom-built for this application, as described in Scheibe et al. (2014). In that work, results of the hybrid multiscale simulation (using STOMP and SPH models) were compare with results from a corresponding single-scale (STOMP only) simulation. The primary difference between the two simulations was that in the hybrid multiscale simulation, reaction rates in the central column of grid cells (Figure 3) are updated at each time step based on the results of pore-scale simulations (SPH), whereas in the single-scale simulation reaction rates remain fixed at the nominal value specified at the beginning of the simulation. A detailed description of model results are given in Scheibe et al. (2014) and is not repeated here. The primary effect of the modified reaction rate in the hybrid multiscale simulation is a more accurate representation of the total mass of reaction product generated. Because of the incorrect assumption of complete mixing at the grid scale in the single-scale model, the effective rate of reaction is too high and the amount of reaction product generated is over-estimated by ~15% percent relative to the hybrid multiscale model. In the current work we are applying our general multiscale framework to the same problem. Rather than custom-built scripts tailored to the specific codes (SPH and STOMP), general scripts are being developed around the JUPITER API that can be straightforwardly applied to any pair of codes (or a single codes applied at two different scales as in use case 2). For use case 1, we will evaluate the results by comparing model predictions to those obtained using the custom scripts developed by Scheibe et al. (2014); results should be the same while the framework will be more general and extensible.

For use case 2, results of the hybrid multiscale simulations will be compared to a single-scale simulation in which grid refinement is used to better capture the structure of the thin mud layer near the river. However, since full refinement will not be feasible, and because of differences in the reaction network needed in the mud layer as compared to the rest of the domain, we expect differences in both the accuracy of the model outputs and the computational efficiency of the solutions.

4 Concluding Remarks

We are developing a general framework that applies a many-task approach to hybrid multiscale coupling of microscale and macroscale porous media flow and reactive transport simulators. The hybrid multiscale approach is relatively new in subsurface hydrology, and is well-suited to the use of high-performance computing and a task parallel script-based simulation environment. Loose coupling of many microscale tasks within a macroscale domain is supported by use of the Swift workflow environment, and provides a feasible solution approach to a complex simulation problem.

The two use cases considered here are relatively simple, and were selected as an initial case for testing our hybrid multiscale modeling framework. Evaluation of the generalized hybrid multiscale framework using these two use cases will provide insights regarding needed improvements and areas of future research. Our long-term goal is to develop a multiscale simulation environment that facilitates the coupling of codes across scales to improve simulation fidelity while maintaining computational efficiency on large parallel systems and minimizing intrusive modifications to the at-scale simulators.

Acknowledgements

This research was supported by the U. S. Department of Energy (DOE) office of Biological and Environmental Research through the Subsurface Biogeochemical Research Scientific Focus Area project at Pacific Northwest National Laboratory (PNNL). Computations described here were performed using computational facilities of the National Energy Research Scientific Computing Center (NERSC), a national scientific user facility sponsored by DOE Office of Science, and computational facilities of the Environmental Molecular Sciences Laboratory (EMSL), a national scientific user facility located at PNNL and sponsored by the DOE Office of Biological and Environmental Research. PNNL is operated for the DOE by Battelle Memorial Institute under Contract No. DE-AC06-76RLO 1830.

References

Banta, E. R., E. P. Poeter, J. E. Doherty and M. C. Hill, (editors) "JUPITER: Joint Universal Parameter Identification and Evaluation of Reliability – An application programming interface (API) for model analysis," Chapter 1 of Book 6. *Modeling Techniques*, Section E. Model Analysis, Denver, Colorado: U. S. Geological Survey, 2006.

Battiato, I., D. M. Tartakovsky, A. M. Tartakovsky, and T. Scheibe, "On breakdown of macroscopic models of mixing-controlled heterogeneous reactions in porous media," *Advances in Water Resources*, 32(11):1663-1673, 2009.

Battiato and D. M. Tartakovsky, "Applicability regimes for macroscopic models of reactive transport in porous media", *J. Contam. Hydrol.*, 120-121: 18-26, doi:10.1016/j.jconhyd.2010.05.005, 2011.

Ben Belgacem, M., B. Chopard, J. Borgdorff, M. Mamonski, K. Rycerz, and D. Harezlak, "Distributed multiscale computations using the MAPPER framework," *Procedia Computer Science* 18: 1106-1115, 2013.

Borgdorff, J., J.-L. Falcone, E. Lorenz, C. Bona-Casas, B. Chopard, and A. G. Hoekstra, "Foundations of distributed multiscale computation: Formalization, specification, and analysis," *J. Parallel Distrib. Comput.*, 73: 465-483, 2013.

Boso, F. and I. Battiato, "Homogenizability conditions for multicomponent reactive transport," Advances in Water Resources 62: 254-265, doi:10.1016/j.advwatres.2013.07.014, 2013.

E, W., B. Engquist, and Z. Y. Huang. "Heterogeneous multiscale method: A general methodology for multiscale modeling," *Physical Review B* 67(9), 092101, 2003.

Falcone, J.-L., B. Chopard and A. Hoekstra, "MML: Towards a Multiscale Modeling Language," *Procedia Computer Science*, 1: 819-826, 2012.

Groen, D., S. J. Zasada, and P. V. Coveney, "Survey of multiscale and multiphysics applications and communities," arXiv:1208.6444v3 [cs.OH], 2013.

Katz, D. S., M. Ripeanu, and M. Wilde, "Many-task computing tools for multiscale modeling," arXiv:1110.0404 [cs.DC], 2011.

Lichtner, P. C., G. E. Hammond, C. Lu, S. Kara, G. Bisht, B. Andre, R. T. Mills, and J. Kumar, *PFLOTRAN User Manual*, 2013.

Nichols, W. E., N. J. Aimo, M. Oostrom, and M. D. White, *STOMP Subsurface Transport Over Multiple Phases: Application Guide PNNL-11216 (UC-2010)*, Pacific Northwest National Laboratory, 1997.

Palmer, B., Gurumoorthi, V., Tartakovsky, A. M., and Scheibe, T. D., "A component-based framework for smoothed particle hydrodynamics simulations of reactive fluid flow in porous media," *Int. J. High Perform. Comput. Appl.* 24: 228-239, doi:10.1177/1094342009358415, 2010.

Poeter, E. P., M. C. Hill, E. R. Banta, S. Mehl and S. Christensen, "UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation," *Techniques and Methods*, Book 6, Ch. A11, Denver, Colorado: U. S. Geological Survey, 283 p., 2005.

Scheibe, T. D., X. Yang, K. Schuchardt, K. Agarwal, J. Chase, B. Palmer and A. Tartakovsky, "A many-task parallel approach for multiscale simulations of subsurface flow and reactive transport," in *Proceedings 7th ACM Workshop on MTAGS*, New Orleans, LA, 2014.

Scheibe, T. D., E. M. Murphy, X. Chen, A. K. Rice, K. C. Carroll, B. J. Palmer, A. M. Tartakovsky, I. Battiato, and B. D. Wood, "An analysis platform for multiscale hydrogeologic modeling with emphasis on hybrid multiscale methods," *Groundwater*, 53(1): 38-56, doi: 10.1111/gwat.12179, 2015

Tang, Y.-H., S. Kudo, X. Bian, Z. Li, and G. E. Karniadakis, "Multiscasle Universal Interface: A concurrent framework for coupling heterogeneous solvers," arXiv:1411.1293v1 [physics.comp-ph, 2014.

Tartakovsky, A., G. Redden, P. C. Lichtner, T. D. Scheibe, and P. Meakin, "Mixing-induced precipitation: Experimental study and multi-scale numerical analysis," *Water Resources Research*, 44, W06S04, doi:10.1029/2006WR005725, 2008.

Tartakovsky, A. M. and T. D. Scheibe, "Dimension reduction method for advection-diffusion-reaction systems," *Advances in Water Resources*, 34(12): 1616-1626, doi:10.1016/j.advwatres.2011.07.011, 2011.

Wilde, M., M.Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster, "Swift: A language for distributed parallel scripting," *Parallel Comput.* 27(9): 633-652. doi:10.1016/j.parco.2011.05.005, 2011

Wood, B. D., "The role of scaling laws in upscaling," *Advances in Water Resources*, 32(5): 723-736, 2009.

Yang, A., "On the common conceptual and computational frameworks for multiscale modeling," *Industrial & Engineering Chemistry Research*, 52: 11451-11462, doi:10.1021/ie303123s, 2013.

Zachara, J. M., P. E. Long, J. Bargar, J. A. Davis, P. Fox, J. K. Fredrickson, M. D. Freshley, A. E. Konopka, C. Liu, J. P. McKinley, M. L. Rockhold, K. H. Williams, and S. B. Yabusaki, "Persistence of uranium groundwater plumes: Contrasting mechanisms at two DOE sites in the groundwater-river interaction zone," *J. of Contaminant Hydrology* 147: 45-72, doi:10.1016/j.jconhyd.2013.02.001, 2013.