Component-Based Framework for Subsurface Simulations

BJ Palmer, Yilin Fang, Glenn Hammond, Vidhya Gurumoorthi

Pacific Northwest National Laboratory, PO Box 999 Richland WA 99352

bruce.palmer@pnl.gov

Abstract. Simulations in the subsurface environment represent a broad range of phenomena covering an equally broad range of scales. Developing modelling capabilities that can integrate models representing different phenomena acting at different scales present formidable challenges both from the algorithmic and computer science perspective. This paper will describe the development of an integrated framework that will be used to combine different models into a single simulation. Initial work has focused on creating two frameworks, one for performing smooth particle hydrodynamics (SPH) simulations of fluid systems, the other for performing grid-based continuum simulations of reactive subsurface flow. The SPH framework is based on a parallel code developed for doing pore scale simulations, the continuum grid-based framework is based on the STOMP (Subsurface Transport Over Multiple Phases) code developed at PNNL. Future work will focus on combining the frameworks together to perform multiscale, multiphysics simulations of reactive subsurface flow.

1. Introduction

Subsurface simulations encompass a rich variety of physical models and numerical techniques and these are embodied in an equally rich variety of both research and production codes. Most of these codes have focused on simulating a single model that usually represents phenomenon at a single scale or at most over a limited range of scales. However, many phenomena of interest in subsurface environments effectively couple a broad range of different scales and represent the interaction of many different processes.

A number of problems immediately crop up when trying to integrate different models together from pre-existing simulation codes. The first and most obvious is that packages that have been developed independently of one another may be written in different languages. More critically, it may be difficult to modify the packages to incorporate the coupling without having to modify large sections of code. Furthermore, subsequent modification and refinement of the combined codes can require considerable additional labor and the whole process may need to begin from scratch if it is decided to add yet another component to the system or to replace one of the existing components with a different model.

This paper will describe the use of the Common Component Architecture (CCA) [1] to develop a framework for performing hybrid subsurface simulations that couple models developed at a number of different scales into a single coherent simulation. Using an object-oriented framework such as CCA for code development has several potential advantages. The deconstruction of existing codes into components and incorporation of the components into the framework helps enforce good software engineering principles by requiring developers to encapsulate different functionality into different components. Furthermore, if the interface has been suitably designed it may be possible to completely replace the functionality represented by the component with an entirely different implementation developed elsewhere. This offers a convenient route to updating codes with the latest software tools or even extending the capabilities of existing codes. Finally, the framework can offer support for automatically managing language interoperability.

The subsurface simulation framework currently under development is focused on decomposing two codes into software components. The first is a smooth particle hydrodynamics (SPH) code designed to simulate flow at the pore scale and the second is a continuum code designed to simulate reactive flows at the Darcy scale. Short term goals include decomposition of the SPH code and abstraction of the communication layer for the SPH code and decomposition of the continuum code and replacement of the grid and solver components with more advanced technology being developed as part of the ITAPS and TOPS projects. The longer term goal is to develop couplings between these simulations and integrate them into the framework as additional components.

2. SPH Component

Smooth particle hydrodynamics (SPH) is a particle based method for simulating the equations of fluid dynamics[2]. However, unlike other particle based techniques, such as molecular dynamics and dissipative particle dynamics, SPH particles do not represent fluid particles directly. Instead, SPH particles represent a random sampling of the continuum hydrodynamic fields. The value of the field at any point can be reconstructed by a suitable convolution of the field values associated with any SPH particle and a suitable weighting function. The particles move over the course of the simulation based on forces that are calculated from the values of the continuum fields. SPH simulations have the advantage that they are relatively easy to implement and can handle a variety of free surface and moving boundary problems with relatively little additional program complexity. SPH simulations are being used in this project to simulate fluid flow and reaction processes at the pore scale.

A typical SPH simulation consists of a loop to integrate the dynamical equations for the particle motion coupled with a force evaluation at each time step. The force evaluation consists of two parts, the first is an evaluation of the density field at each particle location and the second is an evaluation of the hydrodynamic forces. The second evaluation depends explicitly on the density. The density and hydrodynamic force evaluations both require evaluations of interactions between each particle and all the neighbors that lie within a finite interaction distance of the particle. For parallel simulations of large systems, the only scalable way to implement the calculation is to distribute particles among processors such that each processor contains all the particles associated with a given region of space. This approach requires that at each time step, the coordinates of particles from neighboring processors must be collected in a buffer so that interactions between particles on different processors can be included in the calculation. At the end of the calculation, partial sums of the densities and forces for nonlocal particles must also be scattered back to the home processor. Both gathering and scattering particle coordinates and forces requires extensive interprocessor communication.

The componentization of the SPH code focused primarily on creating components for the particle integration routine, the force evaluation, and the communication layer. Numerous other components were also created to manage setup and IO, but in most cases these were straightforward. Because the communication routines in the original SPH code were difficult to modify and were also problem specific, the communication component in the SPH framework was rebuilt from scratch. The new component uses the concept of a data transfer object, which is managed through a data transfer handle that can be assigned an action and associated with an arbitrary number of data vectors. The data vectors represent things like the coordinates and velocities of particles, as well as any other particle attributes the user may want to move around during the transfer. Five actions are currently supported.

They are “shuffle”, “sort”, “update”, “gather” and “scatter_add”. These actions are currently sufficient to account for the particle transfers associated with the force evaluation, plus transfers needed to import and export particles to files.

A complete component “wiring diagram” for an SPH application using the cCaffeine implementation of CCA [1] is shown in the accompanying figure. The force integration loop is in the “Stepper” module, the force evaluation is in the “SPHForce” module, and the communication layer is in the “Lagrange” module. The remaining modules manage the overall calculation and supply input and export functionalities. Also shown are the results of an SPH simulation performed using the framework. The system is a periodic porous medium formed from a cubic lattice of spheres.

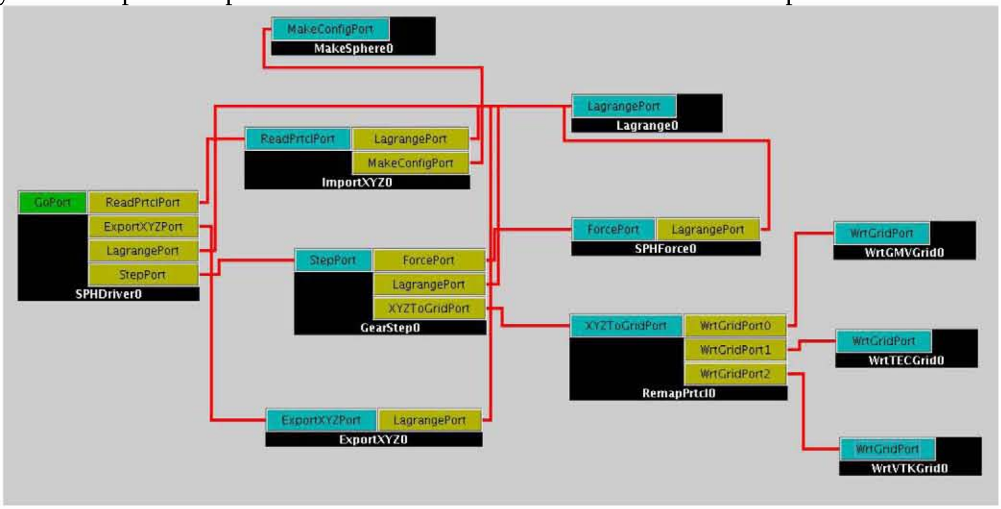

Figure 1. Wiring diagram of SPH application showing relationship between different component modules.

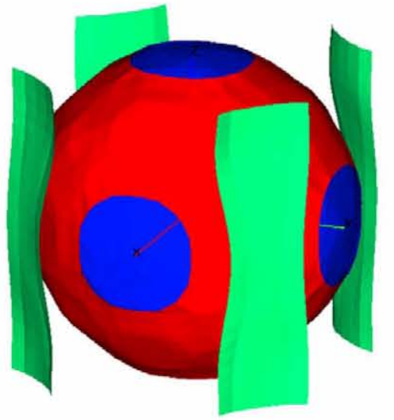

Figure 2. Velocity profile of flow in a periodic cubic lattice of spheres that are in contact with each other. Surface of constant velocity magnitude is shown in green. Flow calculated using the framework shown in Figure 1.

3. STOMP Component

The continuum reactive transport code selected for componentization is the STOMP (Subsurface Transport Over Multiple Phases) simulator[3]. STOMP has been developed by the Pacific Northwest National Laboratory to simulate thermal and hydrological flow and transport phenomena in multiphase or variably saturated subsurface environments. STOMP solves the governing nonlinear mass conservation equations associated with multiphase flow and reactive transport. These nonlinear partial differential equations are discretized spatially on structured orthogonal grids using the integral finite difference approach, and temporally using first-order backward Euler differencing. The nonlinearities resulting from the constitutive equations that relate the primary and secondary variables are solved using Newton-Raphson iteration. The flow equations and the hydrologic transport equations are solved sequentially, while coupled hydrologic transport and mixed geochemical reaction systems are solved using an operator splitting approach (i.e. no iteration between the transport and reaction systems). The figure below illustrates results from STOMP simulations generated for the Hanford SX Tank Farm in Washington State where caustic radioactive wastes are known to have leaked from underground storage tanks. Results of heat transport and variably-saturated flow from two leaking tanks (i.e. SX-109 and SX-108) are shown at left and right, respectively.

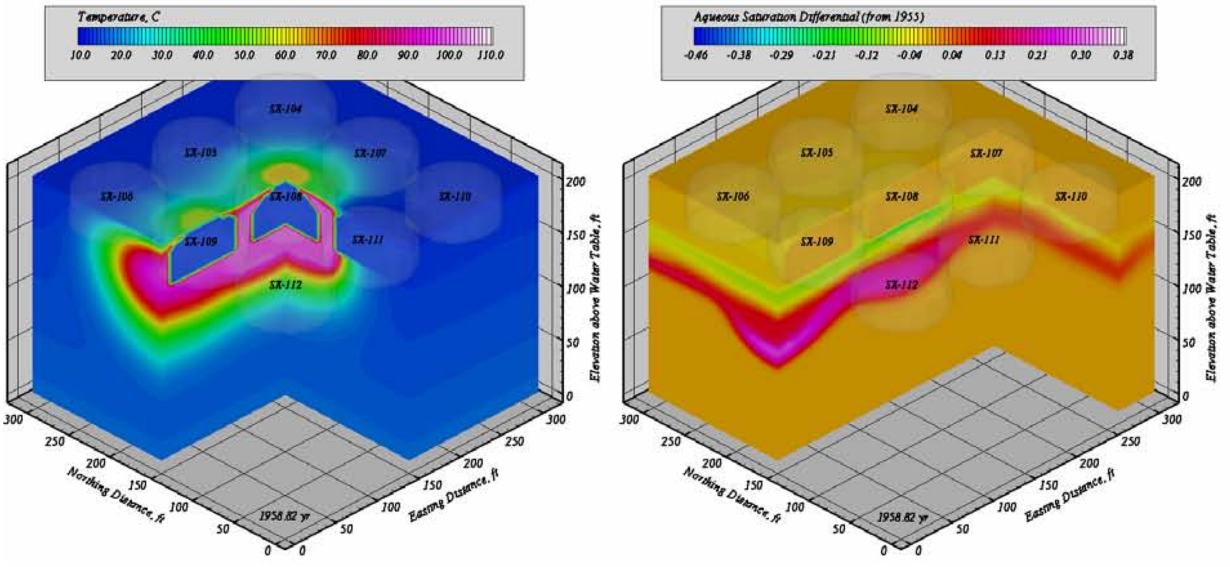

Figure 3. Stomp simulations of heat transport and variably saturated flow in the Hanford SX Tank farm.

The scalable version of STOMP-W (STOMP-W-Sc), which solves the conservation equation for water mass and reactive transport in variably saturated subsurface environments, is being componentized. A solver component that uses the PETSC library[4] has been developed, along with a chemistry component based on the ECKEChem (Equilibrium-Conservation-Kinetic-Equation Chemistry) module[5] in STOMP. These components also use the concept of a data transfer object, which is used to manage data entities such as the discretized matrix equation, the solution vector for the solver component, and the transport properties and species concentration vectors for the chemistry component.

Current efforts are focused on encapsulating the grid functionality into a grid component and replacing the current structured orthogonal grids in STOMP with grids being developed under the Interoperable Technologies for Advanced Petascale Simulations (ITAPS) project[6]. The introduction of new grids into the code may also require more advanced solver technologies. Future efforts will investigate developing more robust and better scaling solver strategies in conjunction with the Towards Optimal Petascale Simulations (TOPS) project, refining the component interfaces for the current STOMP components and further componentizing the remaining STOMP code, and implementing the algorithms to couple the STOMP code to other models, such as the SPH methods described above.

4. Hybrid Model Coupling Interface

Integration of two or more distinct models at fundamentally disparate scales requires 1) determination of spatial subdomains over which each model is to be applied, and 2) definition of algorithms to ensure model consistency at subdomain boundaries. Although very little research has been done on model hybridization in the subsurface sciences[7], a significant body of research exists in other fields from which we can draw[8]. A discussion of hybrid modeling approaches and a brief review of work in other disciplines, as well as an initial example of multiscale model coupling, is given in the companion paper (Scheibe et al., this issue).

The use of a component-based architecture facilitates the hybrid modeling approach by encapsulating the models at each scale, which allows flexible integration of components as well as the potential for changing out which model components are used at each scale. The definitions of the model components and interfaces must accommodate some unique requirements for the hybrid modeling approach and may require additional functionality that would not normally be required for a single-scale modeling approach. Component interfaces must be designed to have flexibility to work with a range of model coupling approaches to be made available to the user as alternatives, and coupling components must be developed for various types of coupling (e.g., particle-particle, grid-grid, and grid-particle).

The Science Application team is currently developing methods for coupling the SPH and STOMP codes. Once these are fully developed and tested they will be instantiated as components that will integrate with the component models currently being developed for the SPH and STOMP codes.

5. Conclusions

A code for performing smooth particle hydrodynamics simulations has been successfully decomposed into a component based framework. Componentization of a continuum subsurface reactive simulation code has proven to be more complicated, but early indications are that this code can be componentized as well. Future work will focus on completing the componentization of the subsurface reactive transport code, refinement of the component interfaces in both codes (to increase flexibility), and development of components that can integrate the two simulation techniques together.

6. Acknowledgements

The work was conducted at the Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RLO 1830. Funding for the project is provided by the Mathematics, Information and Computer Science Division of DOE through the Scientific Discovery through Advanced Computing program.

References

[1] Allan, B.A. *et al.*, *A component architecture for high-performance scientific computing*. The International Journal of High Performance Computing Applications, 2006. **20**: p. 203-231.

[2] Monaghan, J.J., *Smoothed particle hydrodynamics*. Reports on Progress in Physics, 2005. **68**: p. 1703-1759.

[3] White, M.D., M. Oostrom (2000), *STOMP Subsurface Transport Over Multiple Phases, Version 2.0, Theory Guide*. Pacific Northwest National Laboratory, PNNL-12030, UC-2010, Richland, Washington.

[4] Balay S, K Buschelman, V Eijkhout, WD Gropp, D Kaushik, MG Knepley, LC McInnes, BF Smith and H Zhang (2004). *PETSc Users Manual - Revision 2.1.5*, ANL-95/11 Argonne National Laboratory.

[5] White, M. D., B. P. McGrail (2005), *STOMP Subsurface Transport Over Multiple Phases, Version 1.0, Addendum: ECKEChem Equilibrium-Conservation-Kinetic Equation Chemistry and Reactive Transport*, Pacific Northwest National Laboratory, PNNL-15482, Richland, Washington.

[6] http://www.tstt-scidac.org

[7] Balhoff, M.T., K.E. Thompson, and M. Hjortso, *Coupling pore-scale networks to continuum-scale models of porous media.* Computers & Geosciences, 2007. **33**: p. 393-410.

[8] Tartakovsky, D.M. and F.J. Alexander, *Multiphysics modeling - Guest editors' introduction.* Computing in Science & Engineering, 2005. **7**(3): p. 14-15.