# Simulations of subsurface flow and transport at the pore and continuum scales

**Bruce Palmer,[1] Yilin Fang,[1] Tim Scheibe,[1] and Alexandre Tartakovsky[1]**

Pacific Northwest National Laboratory
P.O. Box 999
Richland, WA 99352

E-mail: bruce.palmer@pnl.gov

**Abstract**. This paper presents recent results of simulations of subsurface and flow using both pore scale and grid-based continuum methods. Large scale simulations of pore scale flow and transport using the Smoothed Particle Hydrodynamic method (SPH) using more than 14 million particles have been performed. These simulations are focused on tracking the behavior of contaminants and multiphase fluids in the subsurface. Simulations are also being performed at the field scale using the Subsurface Transport Over Multiple Phases code (STOMP) using a novel, highly-resolved synthetic model of a braided channel belt deposit to analyze the large scale transport of material in a deposit with multiscale heterogeneity. Both sets of simulations are being performed with codes that have been developed using component-based software engineering principles that result in highly modular codes with few dependencies between components.

## 1. Introduction

Groundwater simulations encompass a range of scales, ranging from atomic and molecular scale processes occurring at mineral interfaces to processes at the field scale that can cover tens to hundreds of kilometres. This project has focused on developing methodologies for applying large scale simulations at the pore and continuum scales, with the goal of eventually coupling across these scales using a hybrid simulation approach. The pore scale simulations are based on the Smoothed Particle Hydrodynamics method (SPH), which is a particle based approach to solving the hydrodynamic equations [1]. SPH has the advantage that it can easily be adapted to solving for flow in the topologically complex geometries found in porous media. It can also be modified to handle processes like multi-phase flow. The continuum simulations are based on the Subsurface Transport Over Multiple Phases code (STOMP) developed at Pacific Northwest National Laboratory for performing lab and field scale simulations of subsurface reactive flow and transport [2].

Both simulation codes are being developed using a component-based software engineering approach that is designed to keep the code modular and minimize dependencies between different components. This reduces the overall complexity of the code and simplifies incorporation of new libraries and algorithms. The component-based versions of these codes were originally developed using the Common Component Architecture toolkit (CCA) [3]. The CCA toolkit provides language interoperability [4] and runtime configuration capabilities that are useful in managing multiple components that may have originally been developed using different languages. It also enforces separate namespaces in a way that encourages developers to keep components highly encapsulated.

This is critical for keeping components from developing significant dependencies via shared header files and modules. The communication in both codes is implemented using the Global Arrays toolkit (GA) [6]. The one-sided communication in GA supports a shared memory style programming model that dramatically simplifies much of the parallel portions of the two codes.

## 2. Pore Scale Simulations using the Smoothed Particle Hydrodynamics Method

Smoothed Particle Hydrodynamics is a Lagrangian method for solving the equations describing fluid flow. It is based on the idea that a smooth approximation $A_S$ to a continuum field $A$ can be generated from a discrete sampling of the field in space $A_i$ using the formula

$$A_S(\mathbf{r}) = \sum_i W(\mathbf{r} - \mathbf{r}_i) A_i$$

where $W(\mathbf{r})$ is a normalized weighting function that is at least localized around zero and may be exactly zero outside some finite value of $|\mathbf{r}|$. This equation can be used as the basis for developing approximations to the gradients and higher order derivatives of the continuum field and these, in turn, can be used to write approximate equations of motion for the field. Because the set of points $\mathbf{r}_i$ at which the field is evaluated are carried through the system by local values of the velocity field, they behave like particles and many of the numerical techniques developed for particle simulations can be applied to fluid simulations using the SPH method.

A parallel code based on a spatial decomposition algorithm has been developed as part of this project to simulate the behaviour of millions of particles in 2 and 3 dimensions. The code framework is built around a data manager that provides a mechanism for annotating large data objects in the code so that any component can determine whether it can use or modify a block of data. If a component recognizes a piece of data, it caches a pointer to the data during an initialization step. The data can then be accessed in the remainder of the code using conventional array dereferencing methods, so data access is fast. The advantage of annotating the data is that there is no need to explicitly pass data through argument lists or to share global variables. This simplifies the interfaces of components enormously and eliminates most of the dependencies that mean that a change to one component implies numerous changes to other components. A complete description of the framework is available in [7].

The current SPH framework contains several force components and a simple chemistry component that can be used to simulate transport of a non-reactive contaminant. A wiring diagram for a configuration of the SPH framework using the CCA toolkit is shown in Figure 1.

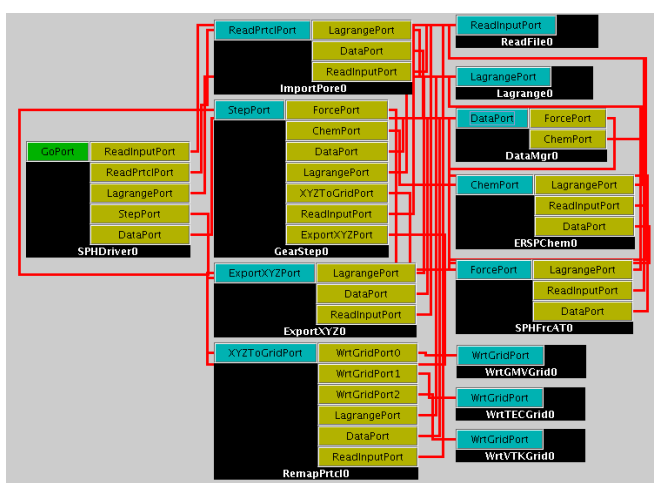


**Figure 1.** Schematic diagram of SPH application built from components.

The existing CCA-based framework is being used for a number of large scale calculations. Some preliminary calculations have been done in two dimensions on an experimental configuration being used to investigate multiphase flow in porous media. The experiment consists of a large array of cylinders between two flat plates. Preliminary numerical tests are in progress to determine if detailed simulations of this system using SPH are feasible. These tests are currently using about 14 million particles on 2048 processes. Plots of the cylinder array test

configuration and the velocity field from an SPH simulation are shown in Figure 2. Additional simulations are planned using terms that account for the thin film nature of the flow.

The SPH code has also been used to investigate multiphase flow in model porous media..A plot of a simulation of two-phase flow in a synthetic porous medium generated through a Monte Carlo algorithm is shown in Figure 3. A non-wetting fluid is entering the bottom of the figure and displacing a wetting fluid. The figure shows fingering of the invading fluid as it tries to displace the wetting fluid coating the solid surfaces.

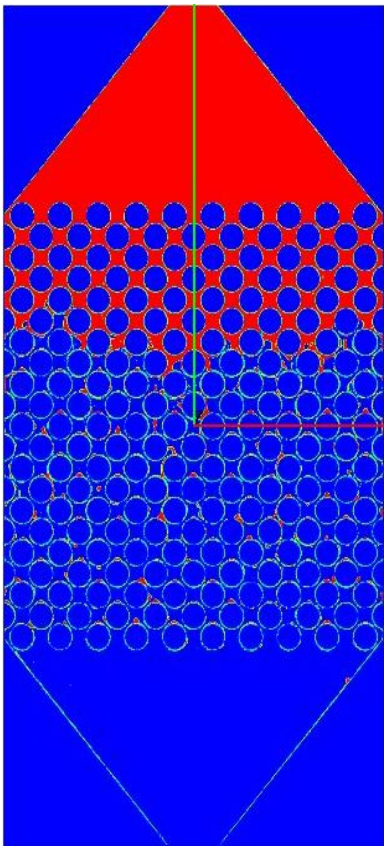

**Figure 2.** Graphic of simulation of a ½ scale experimental configuration of cylinders used to investigate two-phase flow in porous media. The solid material is shown in blue, the displacing fluid is in red.

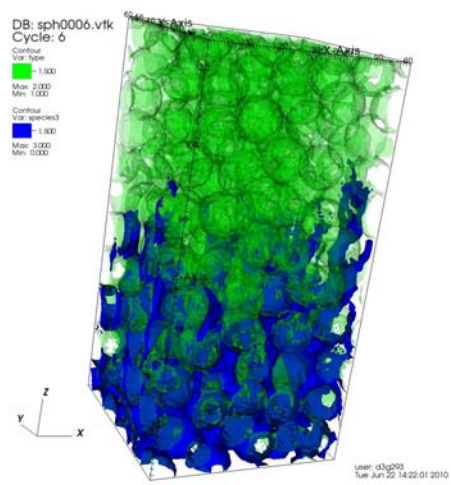


**Figure 3.** Simulation of invasion of a non-wetting fluid (blue) that is displacing a wetting fluid (not shown). The solid matrix representing the porous media is shown in transparent green.

**3. Darcy Scale Simulations using STOMP**

Continuum simulations based on Darcy scale models of flow and transport are used in most subsurface modelling applications. Because of their non-linearity, overall complexity, and tight coupling, these models still present great numerical challenges. Work has been underway to decompose the STOMP subsurface simulator into components using the CCA framework. Many of the strategies used in developing the SPH framework described above are also being employed for STOMP. The decomposition has been based on a data model that treats the numerical grid in STOMP as an unstructured grid. This approach was chosen to enable the future use of unstructured grids. The current

STOMP code has been split into several components. The first is a grid component that initializes and maintains the grid geometry. The grid component can also add and delete fields to the grid and is responsible for updating the ghost cells of fields on the grid when requested to do so by other components. Most of the remaining physics and numerical routines are contained in a second large component, but this has been targeted for further decomposition. Additional components include the initial setup (import), mesh IO (output), and a data manager. The solver routines, chemistry, and time integration routines have also been identified as future targets for componentization. A schematic of the existing STOMP framework is shown in Figure 4.

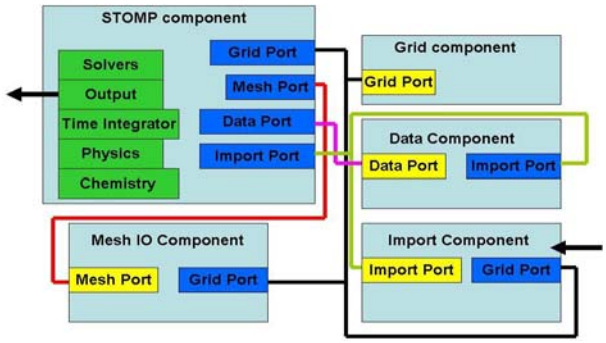


**Figure 4.** Schematic diagram of STOMP framework showing the main physics driver, grid component, mesh IO component, import component, and data manager. Additional potential components are denoted by green boxes. The interfaces defining components are shown by the blue (uses) and yellow (implements) boxes.

The component-based version of STOMP has been used for a number of large simulations. An example is a calculation of flow and transport in a synthetic braided channel belt deposit based on a model by Ritzi [8] that involved 39 million grid cells and was run using up to 4096 processors. A representation of the braided deposit is shown in Figure 5. The flow simulations derived from this geologic structure were subsequently analyzed using particle tracing methods to generate the visualization of the flow field seen in Figure 6. These analyses are being used to determine what procedures can be used to integrate out the fine scale structure from these systems so that larger regions can be simulated using coarser resolution meshes. These upscaling methods are crucial to developing simulation methodologies that can model the behaviour of large regions over hydrologically significant timescales.

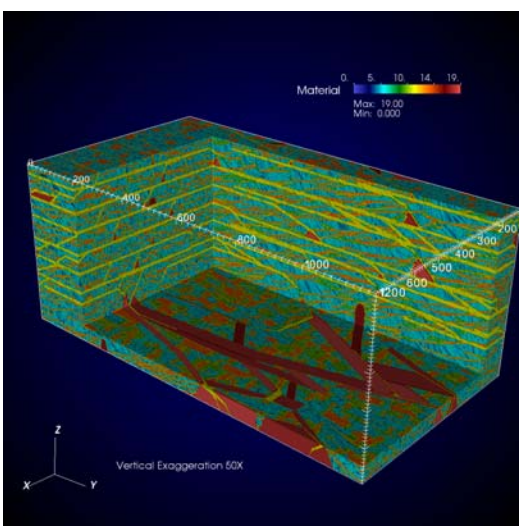


**Figure 5.** Synthetic model of a braided channel belt deposit following Ritzi [8].

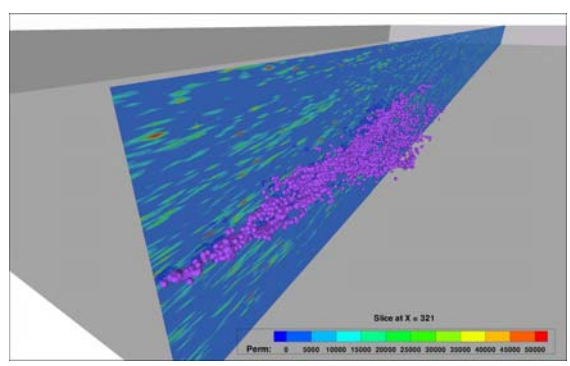

**Figure 6.** Tracers following flow field using the Ritzi configuration shown in Figure 6.

## 4. Summary

The present work has demonstrated that a component-based approach can be used to develop groundwater simulation codes that maintain high performance and can be used to solve a variety of challenging subsurface simulation problems. Further work will focus on systematic studies of flow at the pore scale to develop a detailed picture of the behaviour of two-phase flows in porous media. Simulations of reactive transport in porous media, focusing particularly on uranium transport, are also planned. Continued work on the STOMP code will focus on further decomposition of this code into additional components as well as including additional functionality into the STOMP framework. A primary focus will be to include a $CO_2$ modelling capability that can be used for simulations of carbon sequestration in various geologic formations.


## Acknowledgments

This research is supported by the U. S. Department of Energy's Office of Science under the Scientific Discovery through Advanced Computing (SciDAC) program. A portion of this research was performed using the Molecular Science Computing Facility at EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory.


## References


[1] Monaghan, JJ (2005). *Smoothed particle hydrodynamics Rep. Prog. Phys.*, **68** 1703–1759.

[2] White, MD, and M Oostrom (2000). STOMP Subsurface Transport Over Multiple Phases Version 2.0 Theory Guide, PNNL-12030, 235 pp., Pacific Northwest Natl. Lab., Richland, Wash.

[3] White, MD, and M Oostrom (2006). STOMP Subsurface Transport Over Multiple Phases Version 4.0 User's Guide, PNNL-15782, 120 pp., Pacific Northwest Natl. Lab., Richland, Wash.

[4] Allan BA, R Armstrong, DE Bernholdt, F Bertrand, K Chiu, TL Dahlgren, K Damevski, WR Elwasif, TGW Epperly, M Govindaraju, DS Katz, JA Kohl, M Krishnan, JW Larson, S Lefantzi, MJ Lewis, AD Malony, LC McInnes, J Nieplocha, B Norris, SG Parker, J Ray, S Shende, TL Windus, and SJ Zhou (2006). A Component Architecture for High-Performance Scientific Computing, *Int. J. High Perf. Comput. Applications*, **20** 163–202.

[5] Kohn, S, G Kumfert, J Painter, and C Ribbens, Divorcing Language Dependencies from a Scientific Software Library, 10th SIAM Conference on Parallel Processing, Portsmouth, VA, March 12–14, 2001.

[6] Nieplocha, J, B Palmer, V Tipparaju, M Krishnan, H Trease, and E Apra (2006). Advances, Applications and Performance of the Global Arrays Shared Memory Programming Toolkit, *Int. J. High Perf. Comput. Applications*, **20** 203–231.

[7] Palmer, B, V Gurumoorthi, T Schiebe, and A Tartakovsky (2010). A Component-Based Framework for Smoothed Particle Hydrodynamics Simulations of Reactive Fluid Flow in Porous Media, *Int. J. High Perf. Comput. Applications*, **24** 228–239.

[8] Ramanathan R, A Guin, RW Ritzi, DF Dominic, VL Freedman, TD Scheibe, and IA Lunt (2010). Simulating the Heterogeneity in Braided Channel Belt Deposits: Part 1. A Geometric-Based Methodology and Code, *Water Resources Research*, **46** W04515.