**COUPLING STATE-OF-THE-SCIENCE SUBSURFACE
SIMULATION WITH ADVANCED USER INTERFACE AND
PARALLEL VISUALIZATION**

SBIR Phase I Final Report

Award Number: DE-FG02-07ER84734

Recipient: Thunderhead Engineering Consultants, Inc.

Subcontract: Lawrence Berkeley National Laboratory

Principal Investigator: Brian Hardeman

Final Phase I Report: 6/20/2007 through 3/19/2008

Distribution: Unlimited

Executive Summary

This is a Phase I report on a project to significantly enhance existing subsurface simulation software using leadership-class computing resources, allowing researchers to solve problems with greater speed and accuracy. Subsurface computer simulation is used for monitoring the behavior of contaminants around nuclear waste disposal and storage areas, groundwater flow, environmental remediation, carbon sequestration, methane hydrate production, and geothermal energy reservoir analysis. The Phase I project was a collaborative effort between Thunderhead Engineering (project lead and developers of a commercial pre- and post-processor for the TOUGH2 simulator) and Lawrence Berkeley National Laboratory (developers of the TOUGH2 simulator for subsurface flow). The Phase I project successfully identified the technical approaches to be implemented in Phase II.

Background

This report documents accomplishments of the Phase I SBIR project “Coupling State-of-the-Science Subsurface Simulation with Advanced User Interface and Parallel Visualization” awarded to Thunderhead Engineering Consultants, Inc. The DOE award number is DE-FG02-07ER84734, with Brian Hardeman the Principal Investigator.

The objective of this project is to develop leadership-class computing software for simulation of subsurface transport processes. Phase I is primarily exploratory in preparation for Phase II implementation. Phase I resulted in several discoveries/challenges that will guide Phase II implementation.

The project leverages existing commercial and national laboratory resources to accomplish the technical goals. The project is a collaborative effort between Thunderhead Engineering, which has developed PetraSim (a commercial pre- and post-processor for the TOUGH2 code), and Lawrence Berkeley National Laboratory (LBNL), home of the developers of the TOUGH2 and iTOUGH2 codes. Key personnel that were involved in the original development of the above codes are part of the team, providing the in-depth knowledge of the problem to ensure successful completion of the research.

Applications for the software include: geologic carbon sequestration, environmental remediation and monitoring, transport of radionuclides in the groundwater and vadose zone around nuclear waste disposal and storage areas, methane hydrate production, geothermal reservoir engineering, and general groundwater flow and chemical transport. The developed software will benefit simulating large-scale, application-specific problems of multiphase flow and multicomponent transport processes.

The Phase II research will focus on development of an interoperable set of simulation tools that will support the entire workflow of subsurface modeling. Such an approach is essential when the user is faced with the massive amounts of data associated with large numerical grids, complex hydrogeologic conditions, and complicated transport processes. The research will focus on four key areas:

- An enhanced geometric conceptual model that supports grid-independent, data-driven definition of properties and boundary conditions. Input data will be imported from existing geometric databases and field observations. The model in PetraSim will integrate simulation workflow, from problem definition through parallel execution and visualization.

- The integration of block-structured AMR (see Figure 1 and Figure 2) into TOUGH+ (the object-oriented version of TOUGH2), with corresponding support in PetraSim. The integrated software will efficiently and accurately capture transient evolution of sharp thermal/saturation/concentration fronts in multiphase flow and multi-component transport conditions, and upscale multiple physical processes of flow and transport over multiple scales of naturally heterogeneous media through different mesh refinement levels.
- Integration of iTOUGH2 and iTOUGH+ (the next-generation inverse modeling code based on TOUGH+, currently under development at LBNL), into the PetraSim framework.
- Rapid visualization of the output files resulting from large-scale integrated TOUGH+ and AMR analyses using advanced parallel visualization tools VisIt and ParaView.

Figure 1: Block-structured local mesh refinement in three dimensions (Berger and Oliger, 1984)

Figure 2: The combined AMR and embedded boundary method used to compute the propagation of a supersonic jet into a vacuum. Each box corresponds to a refined grid patch (Colella, 2007).

The result of the project will be new leadership-class computing software for preprocessing, model development, simulation, model calibration, and postprocessing of nonisothermal subsurface flow and transport processes. It will be based on the proven capability of the PetraSim, TOUGH2, and iTOUGH2 codes, and on advances in the AMR algorithms and parallel visualization tools. The new software will be the next generation of these powerful tools.

Technical Approach

The overall approach of the proposed work is to combine available state-of-the-art adaptive mesh refinement technology with the subsurface modeling capability of TOUGH+. In order to make this technology accessible to a broad range of scientists and engineers, the simulation management and geometric modeling tools in PetraSim will be used to generate the high-level conceptual models needed when working with large simulation grids running on multiple processors. This workflow integration is key to successful commercialization of large-scale parallel simulation.

Adaptive Mesh Refinement (AMR)

The overall approach of this research area in the Phase II project is to integrate the advanced AMR technology into our state-of-the-science subsurface simulator TOUGH+. The integrated TOUGH+AMR will significantly enhance our simulation capabilities in tracking various types of dynamic sharp fronts and gradients in an accurate and efficient manner.

The sharp fronts usually result from transport and biogeochemical processes with, for example, intentional release and unintentional spill of waste materials in a natural subsurface system. Mathematically, these processes often exhibit highly hyperbolic features, with advectively moving sharp fronts. Examples in hydrogeological applications include (1) an invading moisture front in unsaturated fractured and porous media, (2) DNAPL migration at a spill site, (3) evolution of a contaminant plume, (4) sharp reaction fronts in biogeochemical remediation, (5) unstable density-driven fingering of dissolved CO₂ and phase viscous fingering in geologic carbon sequestration, and (6) temperature changes associated with evaporation and condensation fronts during reinjection of water into geothermal reservoirs. When these sharp fronts occur in a natural, heterogeneous porous or fractured system with multiple spatial scales, they often transport preferentially through high-permeability channels, leading to sparse, spatially separated mass pools.

While the phenomena of transport with sharp fronts occur commonly under natural conditions, they are especially prevalent in natural systems that have been disturbed by the presence of contaminants or by the processes induced by remediation, which often employs air-sparging, heating, and chemical reaction techniques. Given the increasing efforts of DOE, other government agencies, and industry to clean up contaminated soil and groundwater at many subsurface sites, complex simulations are required to design and optimize remediation activities. Other phenomena important to DOE are viscous CO₂ fingering in heterogeneous, hierarchical sedimentary sandstone and unstable density-driven fingering of dissolved CO₂, which may significantly enhance the capacity and safety of geologic carbon sequestration. Capturing the dynamic fingers of supercritical, liquid, or gaseous CO₂ as well as CO₂ dissolved in the aqueous phase by state-of-the-science AMR techniques can help DOE achieve its goal of greenhouse gas emission reduction for mitigating climate change.

Tracking dynamically evolving flow, transport, and biogeochemical reaction processes with sharp fronts is numerically a great challenge. Static meshes, fixed during the entire simulation time, cannot accurately and efficiently track these dynamic fronts because the fine discretization that is required results in a prohibitively large number of grid cells, even when using parallel computing (Zhang et al., 2003). Static meshes with local mesh refinement are usually generated to honor hydrogeologic layering, subsurface structures and engineered objects, and local-scale heterogeneity and corresponding flow features (see Figure 3). The challenges associated with tracking a dynamically evolving sharp front can be addressed by adaptive mesh refinement

(AMR) technology, whose application to groundwater flow and transport are still very limited (Hornung and Kohn, 1997).

Figure 3: Example (excerpt) of non-uniform, unstructured TOUGH+ mesh used for simulating the fate and transport of radionuclides in fractured rock, concrete, and a passageway in an underground accelerator facility

In the AMR technology, local mesh refinement is used where high-resolution discretization is needed to reduce approximation errors in the regions of sharp fronts. Local mesh refinement needs to adapt to the solutions to track dynamically moving sharp fronts. Mesh coarsening is also a component of the AMR techniques to release the computational resources in the regions where sharp fronts have passed through. In this way, the AMR technique can be used to significantly improve modeling efficiency by reducing the total number of gridblocks, while achieving the same degree of accuracy as a high-resolution fixed mesh. Local mesh refinement and adaptation is guided on the basis of error estimates of numerical approximations, so that accurate simulation of large hydro-thermal-chemical gradients in pressure, saturation, concentration, or temperature is resolved by finer local meshes. Although the structured AMR technique (to be used in this project) uses local uniform grids, representation of irregular model boundaries and embedded physical boundaries can be easily handled using a union of rectangles for geometry and the developed AMR libraries for mass conservation (Collela, 2007).

Geometric Modeling Using Boundary Representation

A high level geometric model of the problem domain becomes a necessity when working with simulation grids that change with time during a simulation. Many existing simulation codes, including the TOUGH family of simulators rely on properties specified on a per-cell basis. However, a purely geometric description of the materials, initial conditions, and boundary conditions

using PetraSim, as is proposed in this research, will allow the cell properties to be determined dynamically during a simulation run as the grid is refined and coarsened.

Boundary representation, a form of geometric modeling where the explicit storage of boundary information describes an object, provides the underlying support for the geometric modeling framework. The boundary of a model consists of adjacency information between point sets known as regions, faces, edges, and vertices. This adjacency information is commonly referred to as topology—in particular, we use non-manifold topology (Baumgart, 1974; Weiler, 1986; Choi, 1989; Rossignac and O'Connor, 1990; Hardeman, 1998). Such an approach organizes the modeling data in a consistent way, independent of the approach to be used for numerical solution.

In topology models, the boundary of a region is comprised of a set of faces, edges and vertices. A typical region boundary would be simply a set of faces. Similarly, faces are bounded by a collection of edges. Every edge is then bound at either end by a vertex. Together, all of the regions, faces, edges, and vertices of the model make up the entire model space. This hierarchical representation of an object is depicted in Figure 4.

Storage of the boundaries of all entities in the model explicitly maintains the adjacency information of the model. Because the adjacencies are stored directly, a boundary representation supports very efficient search and data recovery methods. This simplifies and accelerates many tasks, such as display and local modifications of the model.

Topology may be divided into two major categories—manifold and non-manifold. The geometric modeling framework discussed here utilizes a fully non-manifold topological database. This allows the database to represent some structures, such as a wellbore, with a line, rather than a fully developed 3-D object, supporting more powerful conceptual modeling. The data structure, called the Multi-Link data structure (Hardeman, 1998), is a combination of ideas and techniques from previous work on non-manifold boundary representation (Weiler, 1986; Choi, 1989; Rossignac and O'Connor, 1990).

The use of geometric modeling, combined with advanced visualization and user interface tools, can greatly simplify the task of preparing complex problems for analysis. These advances will allow much broader use of the TOUGH+ and iTOUGH2/iTOUGH+ simulators that have already been developed.

Figure 4: Topological elements in boundary representation modeling

Software Integration and Visualization

The technical approaches proposed in the research also include (1) integrate iTOUGH2/iTOUGH+ into PetraSim to provide hydrogeologists powerful tools for model calibration and sensitivity analysis, which are essential in understanding complex subsurface processes with observations and monitoring, and (2) parallel visualization of modeling and calibration using proved VisIt/ParaView.

Phase I Accomplishments

Phase I was used to lay the foundation for Phase II work. In some areas, such as geometric representation and integration of iTOUGH2 with PetraSim, concepts were implemented in software. In other areas, such as integrating adaptive mesh refinement with TOUGH+ and post-processing, the focus was on exploring appropriate approaches and identifying the most feasible algorithms to be implemented in Phase II.

Geometric Representation of the Problem

Stated Phase I Objectives:

The purpose of Phase I is to demonstrate the feasibility of the technical approach to using multiple geometric formats to handle an irregular geometry encountered in model domain, hydrogeological layers and zones, subsurface openings/objects, etc. The Phase I objectives for the user interface include:

- Demonstration of representing various geometric objects in the subsurface in data structures.
- Development of the capability to sketch irregular boundaries in 3D.

Actual Phase I Accomplishments:

In Phase I we demonstrated the feasibility of representing irregular geometry such as geologic layers in PetraSim. Accomplishing this required both a general geometric representation in PetraSim and a corresponding capability to approximate this representation in the solution grid. The first part was successfully demonstrated by implementing a general triangulated surface in PetraSim. The general triangulated surface makes it possible to represent geologic layers and other geometric features independent of the solution grid.

The Phase I accomplishments addresses the same goals as originally stated, but were implemented somewhat differently. The reason for the change in approach was a result of input from RockWare, the distributor of PetraSim and a company that interacts with a broad range of clients who work on subsurface flow modeling. Instead of duplicating capability already available in programs such as Surfer or RockWorks, we updated PetraSim to import data from those models. We have initially added support for importing contour elevations, scattered XYZ data, and AutoCAD DXF files. Contour files were already a part of PetraSim. The imported XYZ or contour file data points are triangulated using a Delaunay algorithm, and DXF files are defined as triangles and quadrilaterals. After a triangulated surface is imported, it is added as part of the geometry data in PetraSim (see Figure 5).

In the process of providing import support, we found that typical geometry files can be very large, defining surfaces with thousands of triangles. In Figure 5, the imported surfaces contain over 40,000 triangles. We found that the geometry system in place could not add these large triangle counts in a timely manner; the model in Figure 5 originally took overnight to generate. To better handle these files, we augmented the geometry system in PetraSim with hierarchical search trees. We also identified bottlenecks in the geometry's topology code and reworked these areas to significantly improve performance. With these improvements, we were able to generate the model shown in Figure 5 in 1 minute.

Figure 5: Model demonstrating use of complicated surfaces to define geometry

While we significantly improved the capability and performance of the geometry code, there is still some work that must be done to properly handle arbitrary surfaces. Currently each surface is built by adding each triangle individually to the working geometry model. Using this process, we have discovered some limitations in the geometry system caused by ambiguities in describing non-manifold topology. These ambiguities still need to be addressed in order to have a fully-functional geometry system. In addition, because triangles are added individually, many unne-

cessary objects are created to store adjacency information, which can cause the memory requirements of these models to balloon. The example model in Figure 5 is conceptually four surfaces and should be represented this way. However, the Phase I implementation stores this as 40,000+ surfaces—one for each triangle. Representing these properly as only four surfaces with multi-faceted geometry will significantly reduce import time and memory requirements.

Integration of Adaptive Mesh Refinement and TOUGH+

Stated Phase I Objectives:

The Phase I objectives for the implementation of adaptive mesh refinement into TOUGH+ are to:

- Demonstrate the feasibility of integrating AMR into TOUGH+.
- Demonstrate that meshes developed with PetraSim are compatible with the requirements of AMR.

Actual Phase I Accomplishments:

Another of the Phase I objectives was to determine the adaptive mesh refinement (AMR) algorithm best suited to enhance the TOUGH+ simulation capabilities to track sharp fronts of saturation and concentration, etc. The search process included (1) applications of different AMR algorithms in groundwater flow and transport simulations, (2) identification of available AMR algorithms in the fields of computational sciences and fluid dynamics, (3) compatibility of the numerical algorithms in TOUGH+ with AMR algorithms, and (4) the TOUGH+ simulation capacities to be enhanced. These objectives were accomplished during Phase I of this project, as summarized below.

The scientific literature describing applications of AMR to groundwater flow and transport simulations is limited (Mansell et al., 2002). These applications include single-phase flow in heterogeneous porous media (Cao and Kitanidis, 1999), unsaturated flow (Li et al., 2007), multispecies reactive transport (Wolfsberg and Freyberg, 1994), coupled density-dependent flow and transport (Trompert, 1993; van Esch, 2002), and multiphase flow and multicomponent transport (Hornung and Trangenstein, 1997; Pau et al., 2008). All these applications are limited to capability demonstration. To our knowledge, there are no reported applications of integrated flow and transport simulators with AMR to real, large-scale problems in hydrogeology.

In general, two different categories of AMR algorithms have been developed and applied to different fluid dynamics problems. In *single-grid AMR algorithms*, one global mesh with adaptive number and locations of grid nodes is used to account for transient features of simulated processes; these single-grid AMR algorithms are often used with finite element methods, and are relatively easy to implement.

In *composite-grid AMR algorithms*, a sequence of grids of different refinement levels is used; the grid of level 0 covers the entire model domain, while the grids of higher refinement levels cover a portion of the model domain with localized sharp fronts, geometric irregularities, and material discontinuities. Initially, a stationary grid possibly containing multiple refinement levels (to handle geometric irregularity and material discontinuities) is developed, while grids of all refinement levels are updated with the transient evolution of solutions. The mathematical model for each refinement-level grid is identical, except that the values of initial and boundary conditions for higher refinement-level grids change through grid-to-grid communication. This means that numerical solutions for each refinement-level grid are obtained almost independently, except intergrid communication. The intergrid communication includes (1) coarse-to-fine interpolation of

boundary conditions, and (2) mass-conservative update of the coarse-grid solution from the finer-grid solutions by volume averaging. In addition to the intergrid communication, the composite-grid AMR algorithms also include local truncation error estimators or indicators used to determine the grid cells to be flagged for further refinement and regridding of new refinement-level grids.

The block-structured AMR developed at LBNL is a composite-grid AMR algorithm. The LBNL AMR is a mature technology with a variety of implementations and applications for various non-linear combinations of elliptic, parabolic, and hyperbolic partial differential equations (PDEs) (Colella, 2007; Colella et al., 2007a, b). The design of the LBNL AMR library package intends to separate the AMR package from the application-specific integration algorithm of PDEs. The AMR library package includes all classes and algorithms related to the error estimation, regridding, and intergrid communication. In this way, the integration of the LBNL AMR with TOUGH+ would be straightforward and assures the successful completion of the Phase II project. The efficiency the integrated TOUGH+ with AMR will gain mainly depends on the ratio of the volume of the finest grid to volume of the entire model domain using the finest grids, and the accuracy to be achieved depends on the refinement in both space and time. The LBNL block-structured AMR library package provides high-level support for domain decomposition based on assigning rectangular patches to processors, enhancing the parallel implementation and scaling. We believe that the integrated TOUGH+ with AMR will offer efficient parallel computing of large-scale applications.

One additional reason for us to select the LBNL block-structured AMR is upscaling of physical processes in multiscale heterogeneous porous media. Such upscaling is critical for geoscientists to address groundwater flow and transport in natural heterogeneous media. In the block-structured AMR, the solution at a grid cell of the coarsest level is obtained through the volume averaging of the solutions of multiple cells at the finest grid overlapping that coarse-grid cell. Such upscaled solutions are accurate and mass conservative, even though the upscaled material properties need to be addressed separately. The combination of AMR and upscaling will focus our computing resources on the regions with high heterogeneity in material properties and high transient features in physical processes (Niessner and Helmig, 2007).

Integration of iTOUGH2/iTOUGH+ with PetraSim

Stated Phase I Objectives:

The Phase I objectives for the integration of iTOUGH+ and PetraSim are to:

- Design the interface between PetraSim and iTOUGH2.
- Demonstrate the interaction between the two codes using a simplified test case.

Actual Phase I Accomplishments:

The Phase I objectives were to design the interface between PetraSim and iTOUGH2 and to demonstrate the interaction between the codes using a simplified test case. The selected test case was “Problem 1: Sensitivity Analysis, Parameter Estimation, and Uncertainty Analysis” from the published iTOUGH2 Sample Problems (Finsterle, 1999a, b, c). This problem is arranged in 6 parts:

1. Solve the TOUGH2 forward problem,

2. Solve the TOUGH2 forward problem and generate a list of pressures and flow rates for part 3,
3. Sensitivity analysis,
4. Parameter estimation,
5. Uncertainty analysis using FOSM error propagation analysis, and
6. Uncertainty analysis using Monte Carlo simulations.

Work during the first phase of this project primarily focused on supporting the enhanced forward run features (parts 1 and 2) and parameter estimation (part 4). The iTOUGH2 simulator also provides support for parallel execution of multiple TOUGH2 forward runs using PVM (Finsterle, 1998). Supporting this feature within PetraSim was a goal as well. At the completion of Phase I, parts 1 and 2 of this example problem were supported. Support for parallel TOUGH2 forward run execution using PVM was also successfully implemented.

Work to support part 4 of the example problem revealed a limitation in PetraSim's representation of initial conditions. To estimate initial condition parameters iTOUGH2 requires that initial conditions be specified as a global default or by material. When creating the input file for the TOUGH2 forward run, PetraSim specifies initial conditions by cell. Removal of this limitation is scheduled for future work. Support for the other parameter estimation features required by part 4 of the example problem was successfully implemented. Note that parameters that are not initial conditions such as permeability and porosity are fully supported.

Support for parts 3, 5, and 6 of the example problem was largely achieved as a side effect of the implementation of parts 1, 2, and 4, but features required for sensitivity and uncertainty analysis were outside the scope of the first phase.

Figure 6 shows some of the global simulation parameter dialog options that were implemented in Phase I. These dialogs allow users to specify inputs related to the COMPUTATION input hierarchy in iTOUGH2.

Figure 6: The **iTOUGH2 Options** dialogs help users specify global simulation parameters and designate host machines for parallel execution

Figure 7 shows two examples of the user interface that were implemented to support input parameters in iTOUGH2's OBSERVATION hierarchy. The user interface elements for parameter estimation (the PARAMETER input hierarchy) are similar. In both cases the input specification

naturally followed a linear branching and merging pattern that was best implemented with “wizard” user interface elements. This approach was successful and may be adopted in other areas of PetraSim in future works. Parameter and observation data that have been created for a simulation model appear in the tree at the left side of the PetraSim's main view.

Figure 7: The **Add Observation** wizard helps users enter iTOUGH2 options

In some cases, additional features not strictly required by one of the six parts of this example problem, but that were categorically similar to features that were required, were implemented (e.g. objective function estimators, format specification inputs for parameters and observations).

At this time, the iTOUGH2 simulator cannot be executed from within PetraSim's user interface. To run an iTOUGH2 simulation, the user must take the following steps:

1. Create input for the TOUGH2 forward problem and for iTOUGH2 using PetraSim,
2. On the **File** menu, click **Write TOUGH2 File...** to create the input file for the forward problem,
3. On the **File** menu, click **Write iTOUGH2 File...** to create the input file for iTOUGH2, and
4. From the command line, execute iTOUGH2 using the two input files.

Integrated execution of iTOUGH2 is planned for future work.

Visualization of iTOUGH2 results within PetraSim was not an objective during Phase I. Because iTOUGH2 generates output in a form that is already compatible with PetraSim's results visualization system, the implementation details of this portion of iTOUGH2 integration have been reserved for future work.

The iTOUGH2 integration work during Phase I of this work demonstrated that completing support and integration of iTOUGH2 within PetraSim would be low-risk and would be beneficial to the software as a whole. Removing PetraSim's cell-based initial condition limitations would not only enable the remaining portion of iTOUGH2's parameter estimation capability, but would also improve PetraSim support for all supported variants of the TOUGH2 simulator. This improvement would also greatly reduce TOUGH2 input file size and improve readability. Addi-

tional iTOUGH2 inputs needed to complete this work require only localized data and would not require modification of PetraSim's larger data model.

Parallel PostProcessing

Stated Phase I Objectives:

The Phase I objectives for postprocessing include:

- Design of the user interface for postprocessing
- Design of the system architecture to support large data sets

Actual Phase I Accomplishments:

In Phase I, we studied parallel visualization routines provided in the Visualization Tool Kit (VTK) that can be used to quickly generate 3D output data for display. To do this we studied ParaView, a freely available visualization tool that uses these parallel routines. We concluded that utilizing the parallel routines in VTK will be relatively straightforward, but that our biggest challenge was not in displaying large 3D data sets, but in reading the data from simulation results files.

After studying the parallel visualization, we found that there are no specific implementations of parallel routines that are needed to generate the 3D output visualizations in PetraSim, including slices and isosurfaces. These visualizations can be generated in parallel simply by dividing the data and giving each processor in the parallel system a different chunk of data to work on. Most of the difficulty comes from synchronizing the data so the input comes from one shared buffer and the output goes to another shared buffer, which should be fairly simple to overcome.

When we benchmarked the performance of visualizing large 3D output data sets in PetraSim, however, it was clear that generating this 3D output was not our main bottleneck; our bottleneck was file reading and memory utilization. While we did implement some parallel routines to generate the data, this had minimal impact on overall performance because the visualization routines provided by VTK were already very fast. Accordingly, we decided to study efficient ways to read data files and improve memory use.

In the process, we researched programming features that greatly improve our file reading performance and memory usage. The first feature we discovered was memory-mapping files, which is a service provided by the operating system that makes a file appear as a buffer in memory. When a program tries to access this memory, the operating system makes sure that the portion of the file needed is paged into memory, possibly paging out other portions if more memory is needed. This provides an automatic caching system by the operating system, and it means that the entire file does not have to be loaded into memory at once to use the data. It is instead loaded and unloaded dynamically as different portions of the data are requested by the end user of the program over time. In PetraSim when a user looks at results, only a small portion of the output data is viewed, such as the data at one time step or the data at all time steps in one cell, which works very well with the memory-mapping concept. This technique can dramatically reduce memory overhead and file load time because a file can be streamed from disk.

We also investigated the storage of results output data in a binary format. Currently all of the simulators supported by PetraSim use a comma separated value (CSV) file for the result data. These CSV files are stored in an ASCII text format. Text values must be reinterpreted into binary values, which can be slow when there are millions of values. If the files are converted to a

binary format, they will both take much less storage space on disk and be much faster to read because the data is already in a machine-readable format.

In our Phase I project, we demonstrated these two programming approaches. We developed a binary file for use in PetraSim that allows us to jump to any portion of a file we need and quickly extract the necessary data without reinterpreting it from ASCII to binary. To support this binary file, we wrote a converter that, on first opening of a CSV output file, converts and saves the data in a binary file. Subsequent requests for results open the binary file until the results are updated by another simulation. Opening the file only requires a quick search through the file to find time steps and data offsets for each time step. This process typically takes a few milliseconds on a several hundred megabyte file. Once this offset information is extracted, the file is memory-mapped. Results can then be loaded dynamically from the memory-mapped file.

One of the output CSV files we used for testing is approximately 300 MB in size. It originally took over 30 seconds to load into memory – which is the same amount of time required to convert it to a binary file. The binary file loads in 20 milliseconds. The only performance drawback using this approach is loading a portion of the results when that portion is not yet paged into memory. The paging process takes an additional 100 milliseconds.

While we have dramatically improved performance of loading 3D results, some future work may be needed as output grows even further. Right now, our memory mapping approach is limited to mapping individual files less than 2 GB in size. Once users start crossing this threshold in generating output, we will need an approach to virtually map these files. In addition, we would like to modify the TOUGH simulators to directly output a binary file format so that we can eliminate the conversion from the text representation to a binary representation.

By focusing our Phase I efforts on file loading issues, we have greatly improved loading performance and reduced memory overhead when working with large simulator output files.

Summary of Project Activities

Phase I demonstrated that the goals of the project can be accomplished. Specific results of the Phase I work that impacted the Phase II proposal include:

- Increased focus on import of model data from other software, with the deliberate decision to avoid duplication of capability available in other models. The Phase I work on improving import of large data sets from other models will be extended in Phase II.
- The selection of the block-structured adaptive mesh refinement approach for implementation in Phase II.
- The modifications in PetraSim to support basic capabilities of the iTOUGH2 code demonstrate that the Phase II goal of full support for iTOUGH will be accomplished.
- The work done in Phase I to speed post-processing will be directly applicable in Phase II. No significant barriers were identified in Phase I.

In summary, Phase I allowed us to clearly identify the technical approaches to be taken in Phase II and allowed us to explore different implementation approaches for large model support in the user interface.

There is every reason to believe that Phase I has provided a foundation that will ensure the success of Phase II of this project.

Products Developed Under the Award

Because this was a Phase I SBIR project, no completed products were developed under the award. Collaboration with Lawrence Berkeley National Laboratory was integral to the Phase I project and will continue in Phase II.

Computer Modeling

Because this was a Phase I SBIR project, no completed software was developed under the award. Information on PetraSim may be found at <http://www.petrasim.com> and on TOUGH2 at <http://www-esd.lbl.gov/TOUGH2>. At the end of Phase II complete descriptions of the model assumptions and documentation will be provided.

References

- Baumgart, B., 1974, *Geometric Modeling for Computer Vision*, Ph.D. Thesis, Stanford University.
- Berger, M. J., and J. Olinger, 1984, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.*, 53, 484-512.
- Cao, J., and P. K. Kitanidis, 1999, Adaptive-grid simulation of groundwater flow in heterogeneous aquifers, *Adv. in Water Resour.*, 22(7), 681-696.
- Choi, Y., 1989, *Vertex-Based Representation of Non-Manifold Geometric Models*, Ph.D. Thesis, Carnegie-Mellon University.
- Colella, P., 2007, APDEC: algorithms software, *SciDAC Review*, Summer, 22-32 (www.scidacreview.org)
- Colella, P., D. T. Graves, N. D. Keen, T. J. Ligochi, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. van Staalén, 2007a, *Chombo Software Package for AMR Applications, Design Document*, Lawrence Berkeley National Laboratory, Berkeley, CA.
- Colella, P., J. Bell, N. Keen, T. Ligocki, M. Lijewski, B. van Straalen 2007b, Performance and scaling of locally-structured grid methods for partial differential equations, *Journal of Physics: Conference Series 78 (2007) 012013* doi:10.1088/1742-6596/78/1/012013, *SciDAC 2007*.
- Finsterle, S., 1999a, *iTOUGH2 User's Guide*, Report LBNL-40040, Lawrence Berkeley National Laboratory, Berkeley, Calif. (<http://www-esd.lbl.gov/iTOUGH2>).
- Finsterle, S., 1999b, *iTOUGH2 Command Reference*, Report LBNL-40041, Lawrence Berkeley National Laboratory, Berkeley, Calif.
- Finsterle, S., 1999c, *iTOUGH2 Sample Problems*, Report LBNL-40042, Lawrence Berkeley National Laboratory, Berkeley, Calif.
- Finsterle, S., 1998, *Parallelization of iTOUGH2 Using PVM*, Report LBNL-42261, Lawrence Berkeley National Laboratory, Berkeley, Calif.
- Hardeman, B., 1998, *A Geometric Modeling Framework for the Analysis of Fractured Rock Volumes*, M.S. Thesis, Kansas State University.
- Hornung, R. D., and J. A. Trangenstein, 1997, Adaptive mesh refinement and multilevel iterations for flow in a porous media, *J. Comp. Phys.*, 136:522-545.

- Hornung, R. D. and S. R. Kohn, 1997, *Future Directions for Adaptive Mesh Refinement in ASCI and Other LLNL Simulation Projects*, Lawrence Livermore National Laboratory, Livermore, CA.
- Li, H., M. W. Farthing, and C. T. Miller, 2007, Adaptive local discontinuous Galerkin approximation to Richard's equation, *Adv. Water Resour.* 30, 1883–1901.
- Mansell, R. S., L. Ma, L. R. Ahuja, and S. A. Bloom, 2002, Adaptive grid refinement in numerical models for water flow and chemical transport in soil: A review, *Vadose Zone J.* 1, 222–238.
- Niessner, J., and R. Helmig, 2007, Multi-scale modeling of three-phase-three-component processes in heterogeneous porous media, *Adv. Water Resour.* 30, 2309–2325.
- Pau, G. S. H., A. S. Almgren, J. B. Bell, M. J. Lijewski, 2008, *A parallel second-order adaptive mesh algorithm for incompressible flow in porous media*, LBNL Report LBNL-71465, Lawrence Berkeley National Laboratory, Berkeley, CA.
- Rossignac, J. R., and M. A. O'Connor, 1990, SGC: A dimension-independent model for point-sets with internal structures and incomplete boundaries, in *Geometric Modeling for Product Engineering*, Wozny, M. J., J. U. Turner, and K. Preiss (eds.), Amsterdam: North-Holland, pp. 145–180.
- Trompert, R., 1993, Local-uniform-grid refinement and transport in heterogeneous porous media, *Adv. Water Resour.* 16, 293–304.
- United States Environmental Protection Agency (EPA), 2005, Office of Air and Radiation (6608J), EPA 402-F-05-026, www.epa.gov/radiation/yucca.
- van Esch, J. M., 2002, Adaptive multigrid modeling of density-dependent groundwater flow, *the Proceedings of 17th Salt Water Intrusion Meeting*, Delft, The Netherlands, May 6–10, 2002.
- Weiler, K. "Topological Structures for Geometric Modeling." Ph.D. Thesis, Rensselaer Polytechnic Institute, August 1986.
- Wolfsberg, A. W., and D.L. Freyberg, 1994, Efficient simulation of single species and multispecies transport in groundwater with local adaptive mesh refinement, *Water Resour. Res.* 30(11), 2979–2991.
- Zhang, K., Y. S. Wu, and G. S. Bodvarsson, 2003, Parallel computing simulation of fluid flow in the unsaturated zone of Yucca Mountain, Nevada, *Journal of Contaminant Hydrology*, 62–63, 381–399.