

APPLICATION OF PARALLEL COMPUTING TECHNIQUES TO A LARGE-SCALE RESERVOIR SIMULATION

Keni Zhang, Yu-Shu Wu, Chris Ding, and Karsten Pruess

Lawrence Berkeley National Laboratory
1 Cyclotron Road
Berkeley, CA, 94720, USA
e-mail: kzhang@lbl.gov

ABSTRACT

Even with the continual advances made in both computational algorithms and computer hardware used in reservoir modeling studies, large-scale simulation of fluid and heat flow in heterogeneous reservoirs remains a challenge. The problem commonly arises from intensive computational requirement for detailed modeling investigations of real-world reservoirs. This paper presents the application of a massive parallel-computing version of the TOUGH2 code developed for performing large-scale field simulations. As an application example, the parallelized TOUGH2 code is applied to develop a three-dimensional unsaturated-zone numerical model simulating flow of moisture, gas, and heat in the unsaturated zone of Yucca Mountain, Nevada, a potential repository for high-level radioactive waste. The modeling approach employs refined spatial discretization to represent the heterogeneous fractured tuffs of the system, using more than a million 3-D gridblocks. The problem of two-phase flow and heat transfer within the model domain leads to a total of 3,226,566 linear equations to be solved per Newton iteration. The simulation is conducted on a Cray T3E-900, a distributed-memory massively parallel computer. Simulation results indicate that the parallel computing technique, as implemented in the TOUGH2 code, is very efficient. The reliability and accuracy of the model results have been demonstrated by comparing them to those of small-scale (coarse-grid) models. These comparisons show that simulation results obtained with the refined grid provide more detailed predictions of the future flow conditions at the site, aiding in the assessment of proposed repository performance.

INTRODUCTION

The unsaturated zone (UZ) at Yucca Mountain, Nevada, has been investigated extensively as a potential repository site for high-level nuclear waste. Quantitative evaluation of hydrogeologic and geothermal conditions of the site is essential for the

assessment and design of the proposed repository system. A large number of numerical models for different purposes have been developed in recent years. Numerical modeling has played an important role in understanding the hydrogeologic and thermal conditions on the site. Since the early 1990s, several three-dimensional numerical models for the UZ of the Yucca Mountain have been developed. One of the first models was developed by Rockhold et al. (1990). They made a detailed investigation of the UZ in the immediate vicinity of the potential repository. Wittwer et al. (1992, 1995) presented a three-dimensional model that incorporated many geological and hydrological complexities. Ahlers et al. (1995) continued development of the UZ model with increased numerical and spatial resolution. However, the 3-D site-scale UZ flow models, developed in recent years, use in general very coarse numerical grid primarily because of the limitation in computational capacity.

One of the most recent numerical simulations was conducted by Wu et al. (1999). Their model was developed to simulate flow and distribution of moisture, gas, and heat at the Yucca Mountain site for prediction of the current and future hydrological and geothermal conditions in the UZ. Their model took into account the simultaneous flow dynamics of liquid water, vapor, air, and heat in the highly heterogeneous, fractured porous rock in the UZ. Both dual-permeability and effective-continuum modeling (ECM) approaches were used to treat the fracture and matrix system. For both the ECM and dual-permeability conceptual models, the model grid include tens of thousands of gridblocks. For a simulation domain of about 43 km² in area and 700 m in thickness, the spatial resolution used in these model grids was apparently not enough for detailed description of the geological system.

In this paper, the parallel TOUGH2 code is demonstrated for a large-scale simulation. The parallel code was developed from the original TOUGH2 code by introducing the message-passing

interface (MPI) library. The parallel code first partitions an unstructured simulation domain using the METIS graph partitioning programs (Karypsis and Kumar, 1998). The spatially discretized nonlinear equations describing the flow system are then set up for each partitioned part at each time step. These equations are solved with the Newton iteration method. In each Newton step, a nonsymmetric linear equation system is formed for each part of the domain and is then solved using a preconditioned iterative solver selected from the Aztec linear solver package (Tuminaro, et al., 1999). During each Newton iteration, the linearized equation systems must be updated with updating in primary variables. Updating the left-hand side and Jacobian matrix requires communication between different processors for data exchange across the partitioning borders. By distributing the computation time and memory requirements to processors, the parallel TOUGH2 code allows more accurate representation of reservoirs because of its ability to include more detailed information with a refined grid. A one-dimensional verification example is presented to verify the code for applying to a problem with three equations per gridblock by comparing simulation results to the solutions of the original TOUGH2 code.

Parallel computing techniques are applied to the simulation of the UZ at Yucca Mountain through use of the parallel TOUGH2 code. This work is based on the model of Wu et al.(1999), introducing finer gridblocks for the domain spatial discretization. Our 3-D model consists of more than 10^6 gridblocks and 4×10^6 connections (interfaces) to represent the UZ of the highly heterogeneous and fractured tuffs. The simulation was carried out on a Cray T3E-900, a distributed-memory massively parallel computer with 695 processors. Simulation results indicate that the parallel computing technique implemented in TOUGH2 code is very efficient in both computing speed and memory use. The reliability and accuracy of the results have been demonstrated by comparing to those of previous coarse-grid models of single CPU simulations. Results obtained with this refined grid also provide more detailed predictions of the future flow conditions at the site, which will aid in the assessment of proposed repository performance.

DESCRIPTION AND VERIFICATION OF THE PARALLEL CODE

Code Description

TOUGH2 (Pruess et al., 1999) is a general-purpose numerical simulation program for multi-dimensional fluid and heat of multiphase, multicomponent flows in porous and fractured media. The numerical scheme of the TOUGH2 code is based on the integral finite difference (IFD) method (Narasimhan and

Witherspoon, 1976). Conservation equations involve mass of air, water and chemical components as well as thermal energy are discretized in space using the IFD method. Time is discretized fully implicitly using a first-order backward finite-difference scheme. The resulting discretized finite-difference equations for mass and energy balances are nonlinear and are solved simultaneously using the Newton/Raphson iterative scheme. The parallel code was modified from the original version of the TOUGH2 code by introducing the MPI (Message-Passing Interface) message passing library. Other important improvements in the code include optimization of memory use, consideration of the memory requirement balance between all processors, and enhancement of I/O efficiency.

The first step in a parallel TOUGH2 simulation is partitioning of the simulation domain. Efficient and effective methods for partitioning unstructured grid domains are critical for successful parallel computing schemes. Large-scale numerical simulations on parallel computers require the distribution of gridblocks to different processors. This distribution must be carried out such that the number of gridblocks assigned to each PE (processing element) is the same and the number of adjacent blocks for each PEs is minimized. The goal of the first condition is to balance the computation efforts among the PEs; the goal of the second condition is to minimize the time-consuming communication resulting from the placement of adjacent blocks to different processors. In the simulation, a model domain is represented by a set of gridblocks (elements), and the interfaces between every two gridblocks are represented by connections. The entire connection system of gridblocks is defined through input data. From the connection information, an adjacency matrix can be constructed. The adjacency structure of the model meshes is stored with a compressed storage format (CSR). We use three partitioning algorithms implemented in the METIS package version 4.0 (Karypsis and Kumar, 1998). The three algorithms are here denoted the *K-way*, the *VK-way*, and the *recursive* partitioning algorithm. *K-way* is used for partitioning a graph into a large number of partitions (greater than 8). The objective of this algorithm is to minimize the number of edges that straddle different partitions. If a small number of partitions is desired, the *recursive* partitioning method, a recursive bisection algorithm, should be used. *VK-way* is a modification of *K-way* and its objective is to minimize the total communication volume. Both *K-way* and *VK-way* are multilevel partitioning algorithms. Gridblocks are assigned to particular processors through partitioning methods and reordered by each processor to a local ordering. Elements corresponding to these blocks are explicitly stored on the processor and are defined by a set of

indices referred to as the processor's *update* set. The *update* set is further divided into two subsets: *internal* and *border*. Vector elements of the *internal* set are updated using only information on the current processor. The *border* set consists of blocks with at least one edge to a block assigned to another processor. The *border* set includes blocks that would require values from other processors to be updated. The set of blocks not in the current processor, but needed to update components in the *border* set, is referred to as an *external* set.

After domain partitioning, the input data must also be distributed to each associated processor. For a typical, large-scale, three-dimensional model, a memory of several gigabytes is generally required. Therefore, need arises to distribute the memory requirement to all processors. Each processor has a limited space of memory available. To make efficient use of the memory of a processor, the input data files of the TOUGH2 code are organized in sequential format. Two groups of large data blocks reside within a TOUGH2 mesh file: one with dimensions equal to the number of gridblocks, the other with dimensions equal to the number of connections (interfaces). Large data blocks are read one by one through a temporary full-size of array and then distributed to processors one by one. This method avoids storing all input data in a single PE and greatly enhances the I/O efficiency.

Computational efforts are extensive for a large simulation problem. In the parallel code, the assembly and solving of linear equation system is shared by all the processors. After distribution of input data, the discrete mass and energy balance equations for each local part are set up in different processors. These equations are solved using Newton-Raphson iteration method. Jacobian matrices as well as the right-hand sides of the linearized equation systems need to be recalculated at each Newton iteration. Each processor is responsible for computing the rows of the Jacobian matrix that correspond to blocks in the processor's *update* set. Computation of the elements in the Jacobian matrix is performed in two parts. The first part consists of computations relating to individual blocks. Such calculations are carried out using the information stored on current processor and communications to other processors are not necessary. The second part includes all computations relating to the connections. Elements in the *border* set need information from the *external* set, which requires communication between neighbor processors. Before performing these computations, an exchange of relevant variables is required. For the elements corresponding to *border* set blocks, one processor sends these elements to different but related processors, which receive these elements as *external* blocks.

The final, local linear equation systems are solved by different processors using the Aztec linear solver package. We can select different solvers and preconditioners from this package. The available solvers include conjugate gradient, restarted generalized minimal residual, conjugate gradient squared, transposed-free quasi-minimal residual, and bi-conjugate gradient with stabilization methods. Final solutions are derived from all processors and transferred to one processor for output. Results for the connections that cross the boundary of two different processors are obtained by averaging the solutions from the two processors.

Data communication between processors is an essential component of the parallel TOUGH2 code. Although each processor solves the linearized equations of the local blocks independently, communication between neighboring processors is necessary to update and solve the entire equation system. The data exchange between processors is implemented through a subroutine. When this subroutine is called by all processors, an exchange of vector elements corresponding to the *external* set of the gridblocks will be performed. During time stepping or a Newton iteration, an exchange of external variables is required for the vectors containing the secondary variables and the primary variables. Detailed discussion of the implementation of data exchange can be found in Elmroth et al. (1999).

Code Verification

We present a problem designed to examine the accuracy of parallel-version TOUGH2 code in simulating two-phase fluid and heat flow through a fractured medium. The sample problem considers one-dimensional (1-D) vertical flow in the UZ of Yucca Mountain. A single vertical column is extracted directly from the 3-D site scale model (Bodvarsson et al., 1997). The boundary condition specified at the top is a Dirichlet-type condition. The bottom boundary is treated as being at constant gas pressure, liquid saturation, and temperature by specifying a large volume. A liquid water source is introduced to the second element from the top to provide constant water infiltration, with a rate of 3.8652×10^{-8} kg/s.m². The 1-D column is discretized into 26 blocks and 25 connections. Four processors are used to solve this problem. Each processor is responsible for 6 or 7 gridblocks and 6 to 8 connections.

The accuracy of the simulation is evaluated by comparing results to the solutions from the original single-CPU code. The steady-state solutions obtained for this problem are found to be identical by using the original version and parallel version codes. Figures 1 (a) and (b) present a comparison of the liquid

saturation and temperature between the outputs of the parallel and original code. Further verification is presented in next section by comparing measured data with simulation results.

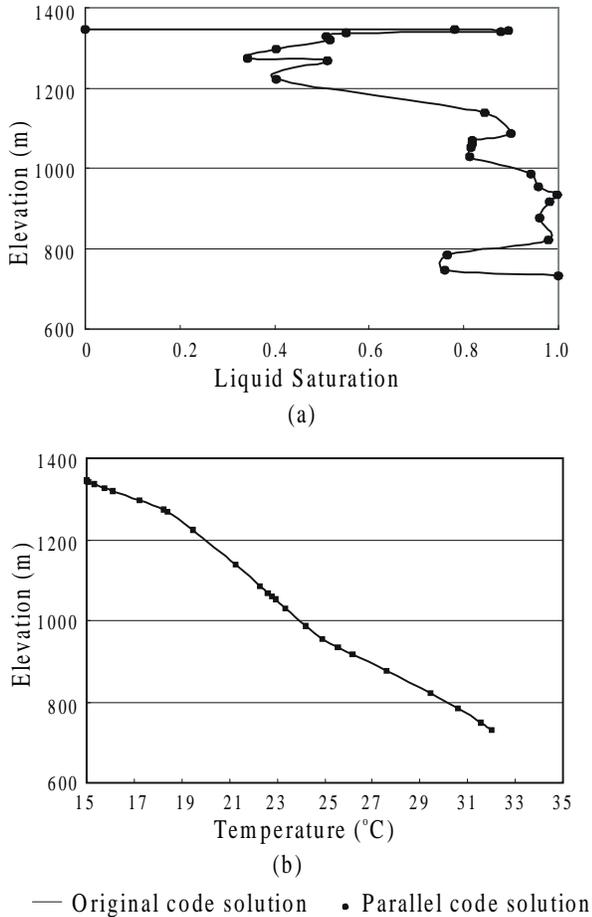


Figure 1. Comparison of the steady-state solutions obtained using original TOUGH2 code and parallel version code: (a) Liquid saturation; (b) Temperature

APPLICATION

This application example is designed to demonstrate the efficiency of parallel computing techniques for multi-component and multiphase large-scale fluid flow simulation. The developed parallel TOUGH2 code is used for the simulation. The simulation results are compared to the solutions of previous coarse grid models. The example is based on a field-scale model developed for investigations of the UZ at Yucca Mountain by Wu et al. (1999). It concerns simulation of flow dynamics of liquid water, vapor, air, and heat in the highly heterogeneous, fractured porous rock in the UZ. Dual-permeability method is used for representing the fracture-matrix system.

The model domain of the UZ at Yucca Mountain covers a total area of approximately 43 km² with the thickness of the UZ varying between 500 and 700 m, depending on the local topography. The proposed repository would be located in the highly fractured Topoph Spring Welded unit, which is located about 300 m above the water table. Geologic formations are organized into hydrogeologic units roughly based on the degree of welding (Montazer and Wilson 1984). From the land surface downwards, we have the Tiva Canyon welded (TCw) hydrogeologic unit, the Paintbrush nonwelded unit (PTn), the Topoah Spring welded (TSw) unit, the Calico Hills nonwelded (CHn), and the Crater Flat undifferentiated (CFu) units. Several strike-slip and normal faults with varying amounts of offset are distributed. The vertical offset along these faults commonly ranges from ten to hundreds of meter, and generally increases from north to south. Detailed discussions of the geologic condition and site characterization of the modeling area can be found in the work of Wu et al. (2000).

The land surface and the water table are taken as the model top and bottom boundaries, respectively. Both top and bottom boundaries are treated as Dirichlet-type. All the lateral boundaries of the domain are considered as no-flow boundaries. The surface infiltration is treated as source terms to the gridblocks in the second grid layer from the top, because the top layer is a first-type boundary with constant temperature, gas pressure, and liquid saturation. Net surface infiltration ranges from 0.02 to 13.4 mm/year (Flint and Flint, 1994) and has an average infiltration rate of 4.6 mm/year over the model domain. The properties used for rock matrix and fractures for the dual permeability model, including two-phase flow parameters of fractures and matrix, were estimated based on field tests and model calibration efforts, as summarized in Wu et al (2000).

The three-dimensional model domain (as well as a three-dimensional irregular numerical grid) used for this study are shown in a plan view in Figure 2. The model grid uses relatively refined gridding in the middle repository area and includes several nearly vertical faults. The grid has about 9,800 blocks per layer for fracture and matrix continua, respectively, and about 60 computational grid layers in the vertical direction, resulting in a total of 1,075,522 gridblocks and 4,047,209 connections. In this model, there are three components for each gridblock, giving 3,226,566 equations per iteration step.

The domain is partitioned into 128 parts by using the K-way partitioning algorithm, with 128 processors used for solving the problem. The ideal case is that the gridblocks can be evenly distributed among the processors with not only approximately the same

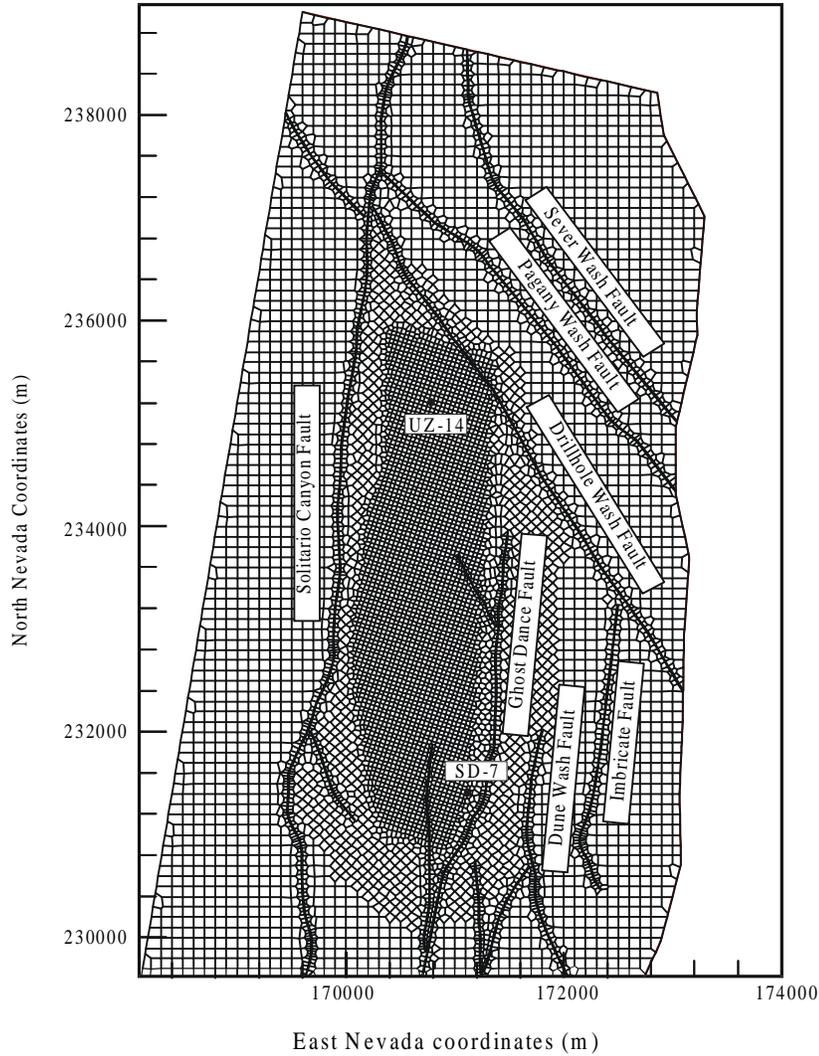


Figure 2. Plan view of the 3-D site-scale model domain, grid, and incorporated major faults

number of internal gridblocks, but also roughly the same number of external blocks per processor. In this problem, the average number of internal blocks is 8,402 at each processor, the maximum number is 8,657, and the minimum number is 8,156. Only about a 6% difference exists between the largest and smallest numbers. The number of external gridblocks is a key parameter for determining the communication volume. In this problem, the average number of external blocks is 2,447, while the maximum number is as large as 3,650 and the minimum as small as 918. This large range indicates that the communication volume can be four times higher for one processor than another. The imbalance in communication volume results in a considerable amount of time wasted on waiting for certain processors to complete their jobs during the solving of equation systems. With this partitioning scheme, the maximum number of connections assigned to a single processor is 37,640. The sum of the

connections at each processor is larger than the total connection number because of the overlapping of connections along the partitioning boundaries. After distribution, the memory requirement for local Jacobian matrix and vectors is only about 20 Mbytes.

The final linear systems are solved using BICGSTAB with Block Jacobian scaling. A domain decomposition based preconditioner (additive Schwarz) with ILUT incomplete LU factorization is selected for preconditioning. The stopping criteria used for the iterative linear solver is

$$\frac{\|r\|_2}{\|b\|_2} \leq 10^{-4}$$

where $\|\cdot\|_2 = \sqrt{(1/n) \sum_{i=1}^n r_i^2}$, n being the total number of unknowns, and r and b the residual and right-hand sides, respectively.

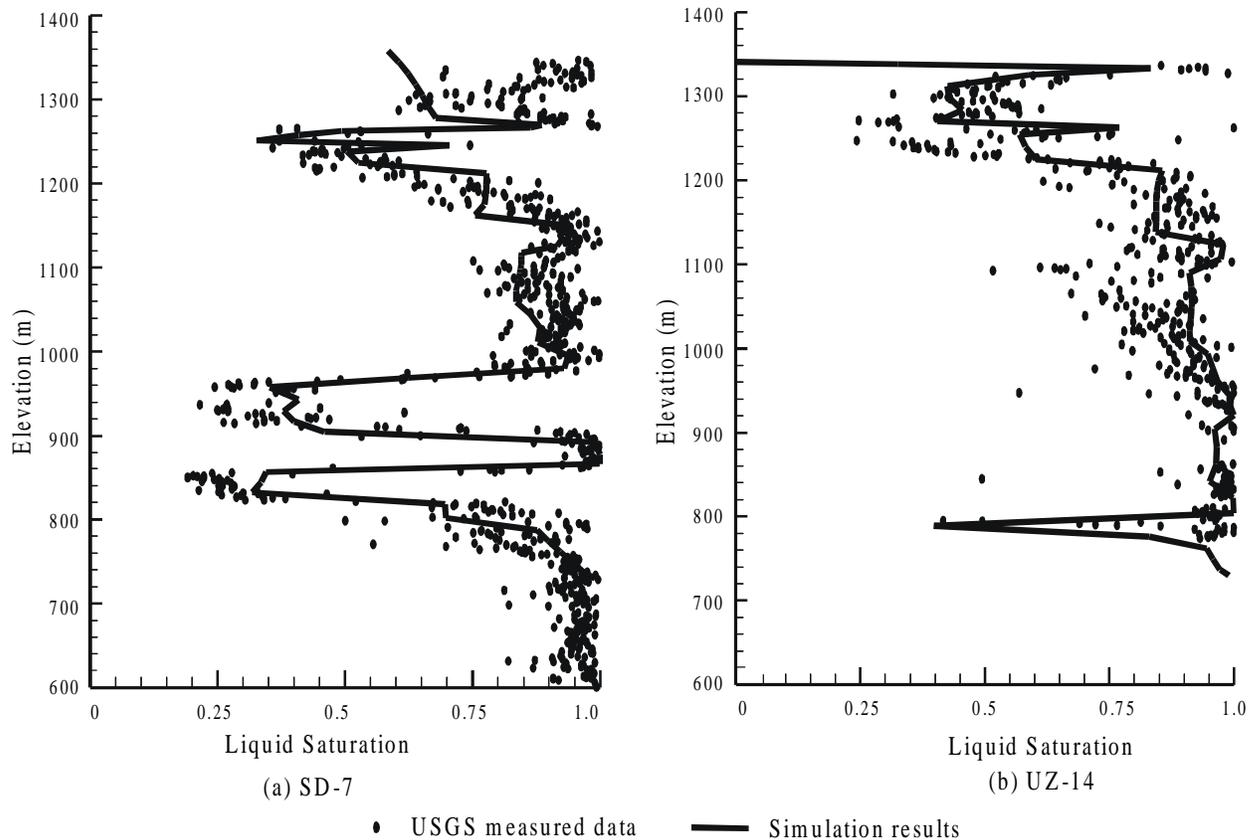


Figure 3. Comparisons of the simulated and observed saturations of borehole SD-7 and UZ-14

Because of the computer batch system time limitations, the production job was allowed to run no more than for 4 hours each session. The simulation runs were set to about 200 steps and 500 steps, which could be completed in less than four hours for isothermal and nonisothermal case, respectively.

Results of the steady-state simulations were used to verify the model. Simulated liquid saturation profiles were compared against measured profiles at two boreholes: UZ-14 and SD-7 (Figure 3a and b). As shown in these figures, the simulated liquid saturations are generally in good agreement with the observed data at the two locations. The liquid percolation flux through the repository horizon and below is one of the most important factors considered in evaluation of repository performance. Figure 4 shows the flux distributions along the repository. The dark color indicates higher values of percolation fluxes with the flux defined in the figures as total mass flux through both fractures and matrix for liquid phase only. Comparison of the simulation results against those using coarse-grid models (Wu, et al., 1999) indicates that the refined-grid model produces results with much higher resolution and more accurate flow distributions at the repository level.

CONCLUSIONS

The parallel TOUGH2 code was introduced and verified for applying to multi-phase and multi-component flow problems through a 1-D example. An application example of the code was presented. Parallel computing techniques implemented in the parallel TOUGH2 code make it possible to develop a 3-D, large-scale multi-phase and multi-component model for simulating moisture and heat flow in the unsaturated zone at Yucca Mountain. Simulation results demonstrate that the parallel TOUGH2 code can efficiently solve multi-phase problems with more than one million gridblocks. Moisture and temperature distributions predicted by the model match with the solutions of former coarse grid models, but with much higher resolution. Simulation results obtained with the refined grid provide detailed predictions of the future flow conditions at the Yucca Mountain.

ACKNOWLEDGEMENT

We thank Lehua Pan for his help in designing the 3-D grid used for the simulation. Thanks are also due to Tianfu Xu and Dan Hawkes for their careful review of this paper. This work was supported, in part, by the Laboratory Directed Research and Development

(LDRD) program of Lawrence Berkeley National Radioactive Waste Management, U. S. Department of Energy, and by the Assistant secretary for Energy Efficiency and Renewable Energy, Office of Geothermal and Wind Technologies, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. This research uses the computation resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U. S. Department of Energy.

REFERENCES

Ahlers, C.F., Bandurraga, T.M., Bodvarsson, G.S., Chen, G., Finsterle, S., and Wu, Y.S. (1995), "Performance analysis of the LBNL/USGS three-dimensional unsaturated zone site-scale model," Yucca Mountain Project Milestone #GLM105M, Lawrence Berkeley National Laboratory, Berkeley, CA.

Bodvarsson, G.S., Bandurraga, T.M., Haukwa, C., Sonnenthal, E.L. and Wu, Y.S. (1997), "The site scale unsaturated zone model of Yucca Mountain for the viability assessment," Yucca Mountain Project Milestone SP24UFM4, Report LBNL-40376, Lawrence Berkeley National Lab, Berkeley, CA.

Elmroth, E., Ding, C., and Wu, Y. (1999), "High performance computations for large scale simulations of subsurface multiphase fluid and heat flow," accepted by *The Journal of Supercomputing*.

Flint, A. L. and Flint, L. E. (1994), "Spatial distribution of potential near surface moisture flux at Yucca Mountain, Nevada," Proceedings of the Fifth Annual International Conference on High Level Radioactive Waste Management, Las Vegas, NV, pp. 2352-2358.

Karypsis, G. and Kumar, V. (1998), "A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, V4.0," Technical Report, Department of Computer Science, University of Minnesota.

Montazer, P., Wilson, W. E. (1984), "Conceptual hydrologic model of flow in the unsaturated zone, Yucca Mountain, Nevada," Water Resources Investigations Report 84-4345, USGS.

Narasimhan, T. N. and Witherspoon P. A. (1976), "An integrated finite difference method for analyzing

Laboratory, and by the Director, Office of Civilian fluid flow in porous media," *Water Resour. Res.*, **12**(1), 57-64.

Pruess, K. Oldenburg, C., and Moridis, G. (1999), "TOUGH2 User's Guide, V2.0," Report LBNL-43134, Lawrence Berkeley National Laboratory, Berkeley, CA, 1999.

Rockhold, M. L., Sagar, B., Connelly, M. P. (1990), "Multi-dimensional modeling of unsaturated flow in the vicinity of exploratory shafts and fault zones at Yucca Mountain, Nevada," Proceedings of the First International Conference on High Level Radioactive Waste Management, Las Vegas, NV, pp. 1192-1199.

Tuminaro, R. S., Heroux, M., Hutchinson, S. A., and Shadid J. N. (1999), "Official Aztec user's guide, Ver 2.1," Massively Parallel Computing Research Laboratory, Sandia National Laboratories, Albuquerque, NM.

Wittwer, C., Bodvarsson, G. S., Chornack, M. P., Flint, A. L., Flint, L. E., Levis, B. D., Spengler, R. W., and Rautman, C. A. (1992), "Design of a 3D site-scale model for the unsaturated zone at Yucca Mountain, Nevada," Proceedings of the Third International High Level Radioactive Waste Management Conference, American Nuclear Society, pp. 263-271.

Witter, C., Chen, G., Bodvarsson, G. S., Chornack, M. P., Flint, A. L., Flint, L. E., Kwicklis, E., and Spengler, R. W. (1995), "Preliminary development of the LBL/USGS three-dimensional site-scale model of Yucca Mountain, Nevada," Report LBL-37356, UC-814, Lawrence Berkeley National Laboratory, Berkeley, CA.

Wu, Y.S., Haukwa, C., and Bodvarsson, G. S. (1999), "A site-scale model for fluid and heat flow in the unsaturated zone of Yucca Mountain, Nevada," *Journal of Contaminant Hydrology*, **38** (1-3), pp.185-217.

Wu, Y. S., Pan, L., Zhang, W., and Bodvarsson, G. S. (2000), "Characterization of flow and transport processes within the Unsaturated zone of Yucca Mountain, Nevada," LBNL Report, Berkeley, CA.

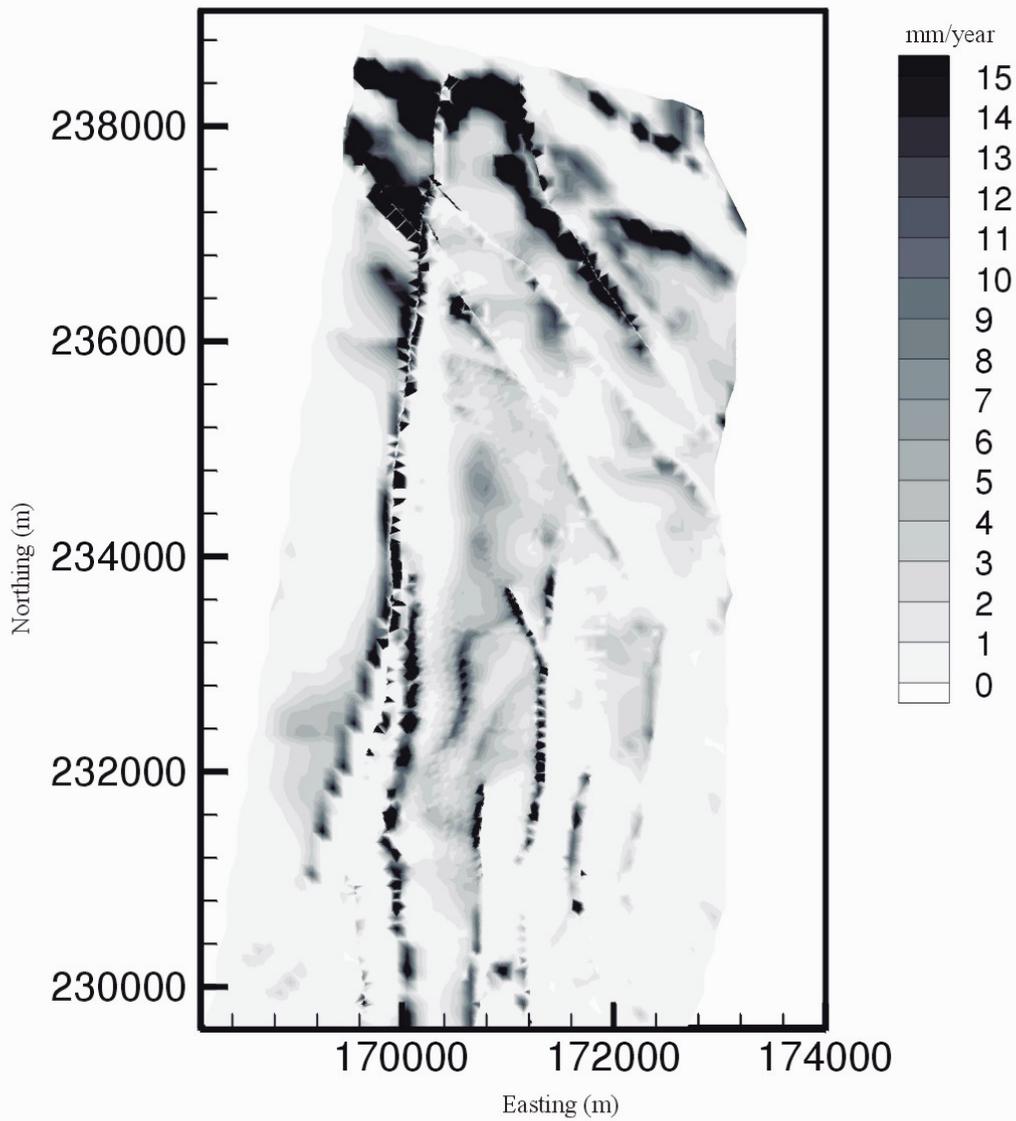


Figure 4. Simulated liquid phase fluxes at repository horizon