Simulating Subsurface Flow and Transport on Ultrascale Computers using PFLOTRAN

Richard Tran Mills[1], Chuan Lu[2], Peter C Lichtner[2], and Glenn E Hammond[3]

[1]Computational Earth Sciences Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6015

[2]Hydrology, Geochemistry, and Geology Group, Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545

[3]Hydrology Group, Environmental Technology Division, Pacific Northwest National Laboratory, Richland, WA 99352

E-mail: rmills@ornl.gov, clu@lanl.gov, lichtner@lanl.gov, glenn.hammond@pnl.gov

Abstract. We describe PFLOTRAN, a recently developed code for modeling multi-phase, multi-component subsurface flow and reactive transport using massively parallel computers. PFLOTRAN is built on top of PETSc, the Portable, Extensible Toolkit for Scientific Computation. Leveraging PETSc has allowed us to develop—with a relatively modest investment in development effort—a code that exhibits excellent performance on the largest-scale supercomputers. Very significant enhancements to the code are planned during our SciDAC-2 project. Here we describe the current state of the code, present an example of its use on Jaguar, the Cray XT3/4 system at Oak Ridge National Laboratory consisting of 11706 dual-core Opteron processor nodes, and briefly outline our future plans for the code.

1. Introduction

Detailed modeling of reactive flows in geologic media is necessary to understand a number of environmental problems of national importance; examples are migration of radionuclides in groundwater and geologic sequestration of CO_2 in deep reservoirs. Such problems generally require three dimensional simulations of non-isothermal systems consisting of multiple phases and possibly dozens of chemical components, and often must incorporate processes operating at different spatial and temporal scales ranging over orders of magnitude. In many cases, detailed understanding will require extremely computationally demanding simulations that will require not only the coming advances in hardware but commensurate advances in algorithms and software.

This paper briefly describes PFLOTRAN, a recently developed and highly scalable code for simulating subsurface flow and reactive transport on machines ranging from laptops to ultrascale computers. We describe the mathematical formulation used by the code, detail its architecture and parallel implementation, present performance results using a benchmark problem from the Nevada Test Site, and briefly outline some of our planned enhancements to the code under SciDAC-2.

2. Mathematical formulation

2.1. Multiphase-Multicomponent Mass and Energy Conservation Equations

The computer code PFLOTRAN solves a coupled system of mass and energy conservation equations for a number of phases including H_2O, supercritical CO_2, black oil, and a gaseous phase. PFLOTRAN

describes coupled thermal-hydrologic-chemical (THC) processes in variably saturated, nonisothermal, porous media in one (1D), two (2D), or three (3D) spatial dimensions. It consists of two modules, PFLOW and PTRAN, which solve for flow and reactive transport, respectively. The multiphase partial differential equations solved by PFLOW for mass and energy conservation can be summarized as [1]:

$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha} s_\alpha \rho_\alpha X_i^\alpha\right) + \boldsymbol{\nabla} \cdot \sum_{\alpha} \left[\boldsymbol{q}_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_\alpha \rho_\alpha \boldsymbol{\nabla} X_i^\alpha\right] \;=\; Q_i^\alpha, \tag{1a}$$

and

$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha} s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T\right) + \boldsymbol{\nabla} \cdot \left[\boldsymbol{q}_\alpha \rho_\alpha H_\alpha - \kappa \boldsymbol{\nabla} T\right] \;=\; Q_e. \tag{1b}$$

In these equations, α designates a phase (e.g. H_2O, supercritical CO_2), species are designated by the subscript i (e.g. $w = H_2O$, $c = CO_2$), ϕ denotes porosity of the geologic formation, s_α denotes the saturation state of the phase, X_i^α denotes the mole fraction of species i; ρ_α, H_α, U_α refer to the molar density, enthalpy, and internal energy of each fluid phase, respectively; $\boldsymbol{q}_\alpha$ denotes the Darcy flow rate defined by

$$\boldsymbol{q}_\alpha \;=\; -\frac{k k_\alpha}{\mu_\alpha} \boldsymbol{\nabla}\big(p_\alpha - W_\alpha \rho_\alpha g \boldsymbol{z}\big), \tag{2}$$

where k refers to the water saturated permeability, k_α denotes the relative permeability, μ_α denotes the fluid viscosity, W_α denotes the formula weight, and g denotes the acceleration of gravity. The source/sink terms, Q_i^α and Q_e, describe injection and extraction at wells, respectively.

The multicomponent reactive transport equations solved by PTRAN have the form [1]:

$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha} s_\alpha \Psi_j^\alpha\right) + \boldsymbol{\nabla} \cdot \sum_{\alpha} \boldsymbol{\Omega}_j^\alpha \;=\; -\sum_{m} \nu_{jm} I_m, \tag{3}$$

for the jth primary species, and

$$\frac{\partial \phi_m}{\partial t} \;=\; \overline{V}_m I_m, \tag{4}$$

for the mth mineral. Ψ_j^α, $\boldsymbol{\Omega}_j^\alpha$ denote the total concentration and flux, defined by the expressions

$$\Psi_j^\alpha \;=\; \delta_{\alpha l} C_j^\alpha + \sum_i \nu_{ji} C_i^\alpha, \tag{5}$$

and

$$\boldsymbol{\Omega}_j^\alpha \;=\; \left(-\tau \phi s_\alpha D_\alpha \boldsymbol{\nabla} + \boldsymbol{q}_\alpha\right) \Psi_j^\alpha, \tag{6}$$

where C_j^α denotes the solute concentration in phase α, and C_i^α denotes the concentration of the ith secondary species related to the concentration of primary species through the mass action equations

$$C_i^\alpha \;=\; (\gamma_i^\alpha)^{-1} K_i^\alpha \prod_j \left(\gamma_j^\alpha C_j^\alpha\right)^{\nu_{ji}^\alpha}, \tag{7}$$

where γ_j^α denotes the activity coefficient and K_i^α the equilibrium constant for reaction (10), below.

The mineral concentration is represented by the volume fraction ϕ_m with molar volume $\overline{V}_m$. The kinetic reaction rate I_m for the mth mineral is assumed to have the form

$$I_m \;=\; -k_m A_m \Phi(\phi_m)\big(1 - K_m Q_m\big), \tag{8}$$

based on transition state theory, where k_m denotes the kinetic rate constant, K_m the equilibrium constant for reaction (11), below; A_m the mineral specific surface area, and Q_m the ion activity product defined by

$$Q_m \;=\; \prod_j \left(\gamma_j^\pi C_j^\pi\right)^{\nu_{jm}}. \tag{9}$$

The factor $\Phi(\phi_m)$ is unity if $\phi_m > 0$ or $K_m Q_m > 1$, and zero otherwise, where ϕ_m denotes the mineral volume fraction. The sign of the rate is positive for precipitation and negative for dissolution and vanishes at equilibrium when $K_m Q_m = 1$.

Chemical reactions included in PTRAN involve homogeneous and heterogeneous reaction between aqueous species and minerals which can be written in the general forms

$$\sum_j \nu_{ji} \mathcal{A}_j \rightleftharpoons \mathcal{A}_i, \tag{10}$$

$$\sum_j \nu_{jm} \mathcal{A}_j \rightleftharpoons \mathcal{M}_m, \tag{11}$$

respectively, where the set of species $\{\mathcal{A}_j\}$ refer to a set of primary or basis species in terms of which all other species are written, $\mathcal{A}_i$ denotes an aqueous complex referred to as a secondary species, and $\mathcal{M}_m$ refers to a mineral. The corresponding thermodynamic equilibrium constants K_i, K_m, and reaction stoichiometric coefficients ν_{ji}, ν_{jm} are derived from an extensive database for aqueous species, gases, and minerals.

2.2. Discretization and numerical solution

In both PFLOW and PTRAN, the governing partial differential equations all have the general form

$$\frac{\partial A}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F} = \mathcal{S}, \tag{12}$$

with accumulation term A, source/sink term $\mathcal{S}$, and flux term $\boldsymbol{F}$ of the form

$$\boldsymbol{F} = \boldsymbol{q}\rho X - \phi D \rho \boldsymbol{\nabla} X, \tag{13}$$

We solve these numerically using an integrated finite volume method. Partitioning the computational domain into a set of finite volumes V_n and integrating the partial differential equations over each volume yields a discretized form of the mass conservation equations. Denoting the kth time step with superscript k, the residual equation for the discretized form of the partial differential equations is

$$R_n = \left(A_n^{k+1} - A_n^k\right) \frac{V_n}{\Delta t} + \sum_{n'} F_{nn'} A_{nn'} - \mathcal{S}_n V_n, \tag{14}$$

The flux $F_{nn'}$ across the $n-n'$ interface connecting volumes V_n and $V_{n'}$ is defined by

$$F_{nn'} = (q\rho)_{nn'} X_{nn'} - (\phi D \rho)_{nn'} \frac{X_n - X_{n'}}{d_n + d_{n'}}, \tag{15}$$

where the subscript nn' indicates that the quantity is evaluated at the interface, and the quantities d_n, $d_{n'}$ denote the distances from the centers of the control volumes V_n, $V_{n'}$ to the their common interface with interfacial area $A_{nn'}$. In general, R_n is a nonlinear function of the independent field variables. We use an inexact Newton method to solve the discretized equations for zero residual.

Within the PFLOW and PTRAN modules the equations are solved fully implicitly, but because PTRAN generally requires much smaller time steps than PFLOW, we couple these modules sequentially. A linear interpolation is used to obtain PFLOW field variables within PTRAN. To account for changes in porosity and permeability due to mineral reactions, PTRAN is used to calculate an updated porosity over a flow time step and the revised porosity is passed back to PFLOW. Future implementations will explore independent grid hierarchies for PFLOW and PTRAN, as well as fully coupled schemes.

3. Architecture and parallel implementation

Both PFLOW and PTRAN employ a modular design and use Fortran 90 features to provide some degree of object orientation. Fortran 90 modules reduce unnecessary sharing of program variables and procedures. PFLOW uses modules and derived types to encapsulate data and the methods that operate on them. A PFLOW driver program loads `pflow_grid_module`, which provides a constructor function `pflowGrid_new` that returns a `pflowGrid` object. That object contains all data about the state of the variables on the flow grid, but the driver does not access those directly. Instead, method subroutines provided by `pflow_grid_module` are used; for example, `pflowGrid_step` advances the flow simulation one time step.

PFLOTRAN has been written from the ground up with parallel scalability in mind, and can run on machines ranging from laptops to the largest massively parallel computer architectures [2, 3, 4]. Scalability has been achieved in large part through the use of the PETSc parallel framework [5] developed at Argonne National Laboratory, which provides tools for managing parallel data distribution and a wide assortment of powerful parallel solvers. PFLOTRAN employs domain-decomposition parallelism. Each processor is assigned a subdomain of the problem and a parallel solve is implemented over all processors. Message passing (3D "halo exchange") is required at the boundary nodes with adjacent processors to fill ghost points in order to compute flux terms. PETSc "Distributed Array" `DA` objects are used to manage the distribution of field variables across the domains and message passing between domains. Ghost point scatters and gathers are handled conveniently by `DA` routines.

Parallel solvers provided by PETSc are used to solve the system of nonlinear equations arising at each timestep. PTRAN uses its own Newton solver routine, while PFLOW uses the `SNES` nonlinear solver framework of PETSc. The `SNES` framework allows easy application of various trust region or line search techniques as well as methods for choosing the accuracy required for each Newton step when an inexact method is employed. Jacobians can be calculated explicitly or, if memory is at a premium, a matrix-free approach, in which the action of the Jacobian is calculated "on-the-fly" via finite-differences, can be employed. However, effective use of this method requires some other approach to preconditioning since a Jacobian matrix is not directly available. Hammond et al. [3] considered physics-based preconditioning based on operator splitting; however, for strong chemical interaction the technique often failed to converge. We note that the modular implementation of the nonlinear solvers we use makes it convenient to add new physics to the code: essentially only a new residual function (and, optionally, a Jacobian calculation routine) for the Newton-Raphson equations must be added. Both PFLOW and PTRAN use the PETSc `KSP` linear solvers and `PC` preconditioners to solve the linear systems arising at each Newton step; this allows the choice of any of several iterative methods (e.g., GMRES, BiCGSTAB, TFQMR) or preconditioners at runtime. Currently we usually use an additive Schwarz method with an incomplete LU factorization applied on each subdomain for parallel preconditioning, but we are adding support for geometric multigrid techniques using the PETSc high-level `DMMG` framework.

4. Modeling the BENHAM test using the Cray XT3/4

To provide an example of the capabilities and parallel performance of PFLOTRAN, we describe an application in which it is being used to understand a case of long distance migration of radionuclides in an area at the Nevada Test Site, where 828 announced underground tests were conducted before the U.S. declared a moratorium in 1992.

The BENHAM test was conducted on December 19, 1968, and had an announced official yield of 1.15 megatons. During the period from 1996 to 1998, a number of radionuclides, including ^{239,240}Pu were detected in trace quantities in two monitoring wells sampling different aquifers at a location 1.3 km from the BENHAM site [6]. Isotopic fingerprinting indicates that the ^{239,240}Pu originates at BENHAM, rather than the much closer TYBO test [7]. This poses a puzzle: How did ^{239,240}Pu migrate to such distant wells located in aquifers that apparently do not interact except through the chimney system created by the test? Fully answering this question will require detailed simulations that couple complex thermal, hydrologic, and chemical processes [8]. We are conducting preliminary investigations using PFLOTRAN to focus on

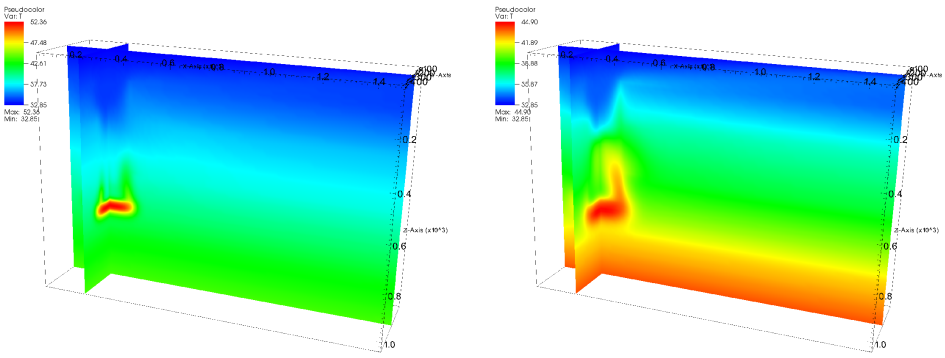

Figure 1. Cross-sections of the simulation domain showing evolution of the temperature distribution (°C) after re-wetting for a simulation with an initial melt glass temperature of 75 °C. The left plot depicts 30 years after re-wetting and the right plot depicts 100 years after re-wetting. Thermal plumes are apparent; radionuclides are carried upward and are then advected by the lateral flow field in the TSA aquifer near the top of the simulation domain.

the complex interplay between heat and fluid flow processes while utilizing a simple non-reactive model of solute transport. These simulations utilize only a limited subset of the capabilities of PFLOTRAN but nonetheless serve to illustrate the scalability of the code on massively parallel machines on a non-trivial problem.

The geometry of the system we must simulate is intrinsically three dimensional. The BENHAM detonation produced a spherical cavity estimated at 200 m in diameter. The intense heat of the blast fused the rock into a "melt glass" at the bottom of the cavity. A cylindrical, rubblized chimney formed as rock above the cavity collapsed, extending from the working point of the test to above the TSA welded tuff aquifer.

Simulations using PFLOTRAN with a relatively coarse grid of $95 \times 50 \times 65 = 308750$ grid points (926250 total degrees of freedom, with pressure, temperature, and solute concentration defined on each node) have explained part of the puzzle by showing a pulse of radionuclides in the TSA aquifer over a narrow window in time due to the creation of convection cells by the heat released from the melt glass [9]. Figure 1 illustrates the development of such convection cells. The grid used in these simulations is fairly coarse, however, and the question of whether simulations utilizing a finer grid will exhibit significantly different evolution of the convection cells arises. We are currently using Jaguar, the Cray XT3/4 system at Oak Ridge National Laboratory consisting of 11706 dual-core Opteron processor nodes, to run much higher-fidelity simulations. Figure 2 shows the parallel scaling we have observed when running on a $256 \times 128 \times 256$ grid (approximately 25 million total degrees of freedom). In this case, we employ BiCGSTAB as the linear solver and use a block-Jacobi preconditioner with block-ILU(0) applied on each Jacobi block. The code scales very well on Jaguar, displaying linear speedup all the way up to 2048 processor cores, and still displaying modest speedup when going from there to 4096 processor cores. This is quite impressive, considering that there are only 6144 unknowns per core when 4096 processor cores are employed.

To further investigate the drop-off in scaling at the highest processor counts, we used the PETSc event logging framework to collect a detailed performance profile. The main culprit appears to be an unexpectedly large jump in the time spent in global dot products (most of which are computed within the BiCGSTAB solver) when going to the largest processor count. As Figure 3 shows, the average time spent in dot products during a flow step jumps from 1.53 seconds at 2048 processor cores to 1.84 seconds at 4096 cores! This jump, in fact, accounts for almost all of the departure from linear scaling observed when going from 2048 to 4096 cores: the wall-clock time consumed by all non-dot product operations is approximately halved when going from 2048 to 4096 cores. A natural question to ask is whether

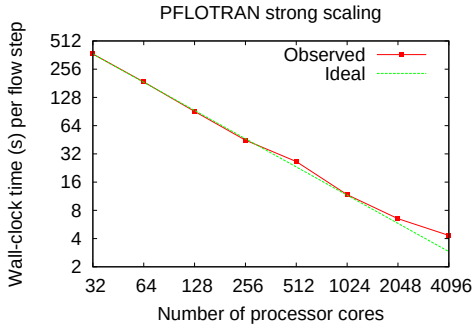

Figure 2. Parallel strong scaling observed for a 25 million degrees of freedom benchmark simulation on Jaguar, the Cray XT3/4 system at ORNL.

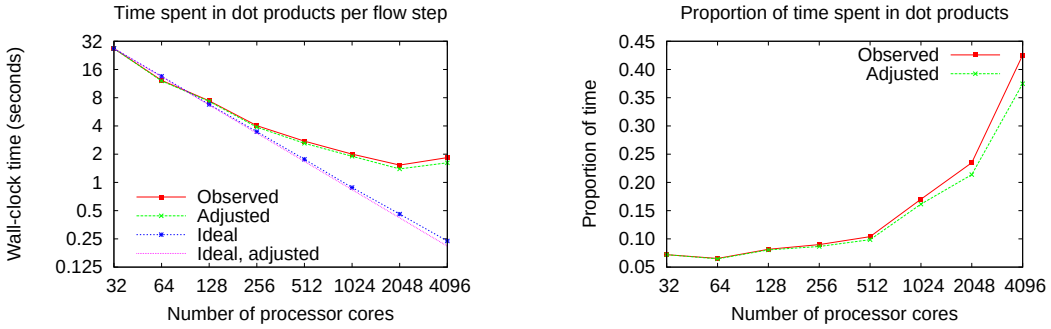

Figure 3. Trends in the time spent in dot products per flow step for the 25 million degrees of freedom benchmark on Jaguar. The left plot displays the average time spent in dot products per flow step, while the right one depicts the proportion of total wall-clock time spent in dot products. Because the nature of the preconditioner employed causes the number of BiCGSTAB iterations (and hence the number of dot products computed) to grow with increasing number of processor cores employed, we include "adjusted" curves that depict what the trends would be if the number of dot products remained constant.

our choice of preconditioner is partly responsible for the increased time spent in dot products. Because the number of Jacobi blocks is equal to the number of processor cores, as processor count increases the blocks become smaller and the preconditioner less effective; as Figure 4 illustrates, more BiCGSTAB iterations and, consequently, more dot products (4 per BiCGSTAB iteration) will be required. The decrease in the effectiveness of the preconditioner, however, is quite mild—at 2048 processor cores, an average of 3493 dot products are computed over one flow step, and at 4096 processor cores, 3610 dot products are computed. In Figure 3, we plot "adjusted" curves based on the observed cost of individual dot products, but assuming that the number of dot products remains constant as the number of processor cores increases; that is, we effectively subtract out the trend due to our choice of preconditioner to obtain trends that are correlated solely with the cost of a global dot product operation as processor core count increases. These trends clearly indicate that the departure from linear scaling that we have observed is largely connected to a similar trend in the performance of global dot products. Because this performance is almost entirely tied to that of `MPI_Allreduce` collectives, it may be possible for us to improve scalability by better tuning these collectives for large processor counts. Clearly, however, we should also experiment with more powerful preconditioners (of which PETSc already provides several) and solvers that will allow us to reduce the number of dot products required.

5. Conclusions and future directions

PFLOTRAN has proved to be a highly scalable code for the simulation of multiphase, multicomponent subsurface reactive flows on machines ranging from laptops to leadership-class supercomputers. It is

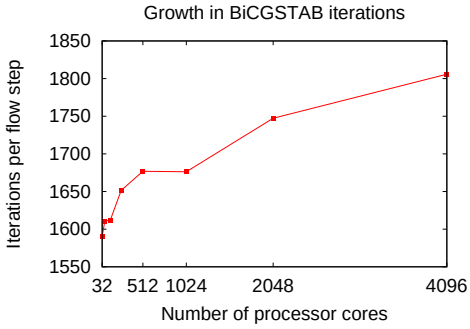

Figure 4. The number of BiCGSTAB iterations per flow step as a function of the number of processor cores used.

currently restricted to conventional single-continuum models with simple orthogonal grids, however, and this limits its predictive capability for problems posessing an intrinsically multiscale nature. Significant enhancements will be made during our SciDAC-2 project to extend its usefulness for multiscale problems. In order to capture sharp reaction fronts and evolving phase boundaries, efforts are currently underway to add adaptive mesh refinement (AMR) capability. Although many communities have been using AMR for years, utilizing it for subsurface flow presents new challenges, because the governing equations involve many scale-dependent quantities such as permeability, dispersivity, and reactive surface area; the upscaling techniques required to transfer these quantities between different refinement levels must be carefully considered. Below the scales that AMR can resolve, we are also adding sub-grid models utilizing multiple interacting continua to capture fracture-matrix interactions, inter-grain diffusion, and dead-end pores. In order to effectively capture irregular problem geometries, we are also adding unstructured grid capabilities using PETSc data structures based on topological sieves. Finally, we are also extending the capabilities of PETSc by developing solvers specifically tailored for generic multi-phase simulations.

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. Computing time was provided by a 2007 DOE INCITE award for the project "Modeling Reactive Flows in Porous Media." This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Initial development of the PFLOTRAN code was supported in part by two DOE Computational Science Graduate Fellowships, administered by Krell Institute.

References

[1] Lichtner P C, Steefel C I and Oelkers E H *Reactive transport in porous media* Reviews in Mineralogy **34**
[2] Mills R T, Lichtner P C and Lu C 2005 *SuperComputing (SC'05) poster abstract* (Seattle, WA)
[3] Hammond G E, Valocchi A J and Lichtner P C 2005 *Advances in Water Resources* **28** 359–376
[4] Lu C and Lichtner P C 2005 PFLOTRAN: massively parallel 3-D simulator for CO_2 sequestration in geologic media *Fourth annual conference on carbon capture and sequestration DOE/NETL*
[5] Balay S, Eijkhout V, Gropp W D, McInnes L C and Smith B F 1997 *Modern Software Tools in Scientific Computing* ed Arge E, Bruaset A M and Langtangen H P (Birkhäuser Press) pp 163–202
[6] Wolfsberg A, Glascoe L, Lu G, Olson A, Lichtner P, McGraw M, Cherry T and Roemer G 2002 *Tech. Rep. LA-13977 Los Alamos National Laboratory*
[7] Kersting A B, Efurd D W, Finnegan D L, Rokop D J, Smith D K and Thompson J L 1999 *Nature* **397** 56–59
[8] Tompson A F B, Bruton C J, Pawloski B A, Smith D, Bourcier W, Shumaker D, Kersting A, Carle S and Maxwell R 2002 *Environmental geology* **42** 235–247
[9] Lichtner P C and Wolfsberg A 2004 Modeling thermal-hydrologic-chemic (THC) coupled processes with application to underground nuclear tests at the Nevada Test Site: a "Grand Challenge" supercomputing problem *Tech. Rep. LA-UR-04-1826*