NONISOTHERMAL MULTIPHASE SUBSURFACE TRANSPORT ON PARALLEL COMPUTERS

M. J. Martinez, P. L Hopkins and J. N Shadid
Sandia National Laboratories
P. O. Box 5800, MS0834
Albuquerque, New Mexico, 87185-0835
E-mail: mjmarti@sandia.gov

ABSTRACT

We present a numerical method for nonisothermal, multiphase subsurface transport in heterogeneous porous media. The mathematical model considers nonisothermal two-phase (liquid/gas) flow, including capillary pressure effects, binary diffusion in the gas phase, conductive, latent, and sensible heat transport. The Galerkin finite element method is used for spatial discretization, and temporal integration is accomplished via a predictor/corrector scheme. Message-passing and domain decomposition techniques are used for implementing a scalable algorithm for distributed memory parallel computers. An illustrative application is shown to demonstrate capabilities and performance.

1. INTRODUCTION

Many research activities in subsurface transport require the numerical simulation of multiphase flow in porous media. This capability is critical to research activities in environmental remediation (e.g. DNAPL contaminations), nuclear waste management, reservoir engineering, and to the assessment of the future availability of groundwater in many parts of the world. Scientific advancements in each of these areas could benefit from a high-performance numerical simulation capability. This paper presents an unstructured grid numerical algorithm, developed under the laboratory-directed research and development (LDRD) program at Sandia National Laboratories (SNL), for subsurface transport in heterogeneous porous media implemented for use on massively parallel (MP) computers via message-passing and domain decomposition techniques. Among the primary objectives of this research were to investigate the use of MP computing for general multiphase systems (which involve such complications as phase appearance and disappearance), and in particular, to develop scalable algorithms for general unstructured grids. The numerical platform which is presented provides an excellent base for new and continuing research in areas of current interest to the geoscience community.

2. MATHEMATICAL FORMULATION

The governing equations for nonisothermal multiphase flow in porous media are statements of mass balance of water and air, over both liquid and gas phases, and a statement of energy balance, also over both phases (e.g., Peaceman, 1977; Martinez, 1995). The canonical form of this coupled system of balance equations is given by

$$\frac{\partial}{\partial t}\begin{bmatrix} d_w \\ d_a \\ e \end{bmatrix} + \nabla\bullet\begin{bmatrix} F_w \\ F_a \\ q \end{bmatrix} = \begin{bmatrix} Q_w \\ Q_a \\ Q_e \end{bmatrix}, \tag{1}$$

where subscripts w, a and e denote water, air and energy. The bulk mass densities are given by,

$$d_\alpha = \phi \sum_{\beta = l, g} Y_{\alpha\beta}\rho_\beta S_\beta, \qquad \alpha = w, a \tag{2}$$

and the energy density is given by,

$$e = (1-\phi)\rho_s e_s + \phi(S_l\rho_l e_l + S_g\rho_g e_g). \tag{3}$$

In these and subsequent equations subscripts l, g, and s refer to the liquid, gaseous, and solid phases, respectively. Also, ϕ denotes porosity, $Y_{\alpha\beta}$ is the mass fraction of component α in phase β, ρ is phase density, e is internal energy, and S_β is phase

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

saturation, and the pore space is assumed fully occupied by the liquid and gas phases, $S_l + S_g = 1$. The net component mass fluxes are defined as

$$F_\alpha = Y_{\alpha l}\rho_l v_l + Y_{\alpha g}\rho_g v_g - \rho_g D_{\alpha g}\nabla Y_{\alpha g}, \quad (4)$$

where v_β denotes the Darcy flux vector and the effective binary diffusion coefficient is given by (Vargaftik, 1975; Pruess, 1987),

$$D_{\alpha g} = \frac{\phi S_g}{\tau} D^0_{\alpha g} \frac{P_{ref}}{P_g}\left(\frac{T}{T_{ref}}\right)^{\upsilon} \quad (5)$$

in which τ is tortuosity, $D^0_{\alpha g}$ is the free-space binary diffusion coefficient between water vapor and air, P is pressure, T is temperature, and υ is roughly 2 for air and water. The advective flux of air dissolved in water is ignored in the present implementation.

In describing flow through porous media, one must necessarily pose average quantities (defined with respect to a representative elementary volume (REV), Bear, 1972) rather than pore-scale continuum values. For example, even though the Darcy fluxes, v_β, have units of velocity, they represent the local volume flux per unit area of porous medium. As such, they cannot satisfy the no-slip condition on fluid/solid interfaces. The flux relations to follow take the place of the momentum balance in continuum equations; they are the volume-averaged momentum balances under conditions of "creeping flow," i.e., inertia-free, slow viscous flow (e.g., Martinez and Udell, 1990).

The advective fluxes are assumed to be properly described by the extended Darcy law, in which relative permeabilities are introduced to account for the multiphase motion of fluids. Thus the Darcy flux vector of phase β (liquid or gas) is,

$$v_\beta = -\frac{k_{r\beta}}{\mu_\beta} k \bullet (\nabla P_\beta + \rho_\beta g) \quad (6)$$

where P is pressure, g is the gravitational acceleration vector, and μ is dynamic viscosity. The intrinsic permeability tensor of the medium is k and the relative permeabilities are denoted $k_{r\beta}$. A particular form of the relative permeability function is described by Eq. (24), to be discussed in the following. The intrinsic permeability tensor is assumed to be a property of the material under consideration, and as such is a spatially heterogeneous quantity. Note that we have assumed that each phase has its own phase pressure. Because the transport problem under consideration involves multiple, immiscible phases, capillary forces must be considered. The phase pressures are related via the capillary pressure relation,

$$P_g - P_l = P_c(S_l) \quad (7)$$

which, as indicated, is assumed to be empirically specified as a function of the phase saturation of liquid. One of several possible models is presented in the application section.

The net heat flux vector is defined by

$$q = -\lambda_T \nabla T + \sum_\beta \rho_\beta v_\beta h_\beta + \sum_\alpha h_{\alpha g} J_{\alpha g} \quad (8)$$

where λ_T is a saturation-dependent effective conductivity, h_β is phase enthalpy, $h_{\alpha g}$ is the enthalpy of component α in the gas phase, and $J_{\alpha g}$ is the gas-phase diffusion flux, the last term on the right-hand-side of Eq. (4).

The foregoing describes the major components of the mathematical model. Some additional information concerning thermodynamics and transport models can be found in Martinez *et al.*, 1997.

3. NUMERICAL METHODS

The numerical method presented for solving the initial-boundary value problem formed by the coupled system of governing equations is a finite element method (FEM), enabling a general representation of complex geologic stratigraphy. The spatial discretization is accomplished by the Galerkin finite element method (e.g. Hughes, 1987). The resulting system of ordinary differential equations is integrated forward in time by a variable-step backward-difference predictor-corrector scheme. The backward-difference formulae result in systems of nonlinear algebraic equations to be solved for nodal quantities. The effective treatment of the highly nonlinear system considered here is made possible by the use of Newton iteration.

3.1 Spatial Discretization

Derivations of the Galerkin finite element method (GFEM) are widely available (e.g., Huebner and Thornton, 1982; Hughes, 1987) and only an abbreviated discussion is presented here. The idea is to approximate the unknown variables appearing in the governing equations by a finite-dimensional set of basis functions with compact support (each basis function is non-zero in a small subdomain of the complete computational domain). The FEM also involves subdividing the computational domain into an assemblage of subdomains, the finite elements. This representation by a finite-dimensional basis is approximate, and results in some error, called a residual. In the general method of weighted residuals, this residual is required to be zero in a weighted-average sense by forming the so-called weak-form residual equation. The weak-form residual equations for the multiphase system are defined by

$$\begin{bmatrix} R_{w,I} \\ R_{a,I} \\ R_{e,I} \end{bmatrix} = \int_{\Omega} \Phi_I \sum_J \Phi_J \begin{bmatrix} \dot{d}_{w,J} \\ \dot{d}_{a,J} \\ \dot{e}_J \end{bmatrix} d\Omega - \int_{\Omega} \nabla \Phi_I \bullet \begin{bmatrix} \boldsymbol{F}_w \\ \boldsymbol{F}_a \\ \boldsymbol{q} \end{bmatrix} d\Omega$$

$$- \int_{\Omega} \Phi_I \begin{bmatrix} Q_w \\ Q_a \\ Q_e \end{bmatrix} d\Omega + \int_{\Gamma} \Phi_I \begin{bmatrix} \boldsymbol{F}_w \bullet \boldsymbol{n} \\ \boldsymbol{F}_a \bullet \boldsymbol{n} \\ \boldsymbol{q} \bullet \boldsymbol{n} \end{bmatrix} d\Gamma \qquad (9)$$

where $I = 1, \ldots, N$, and N is the number of weight functions, Φ_I, $\dot{d}_{\alpha,J}$ is the time-rate of change of the bulk mass density at nodal point *J*, and $\dot{e}_J$ is the time-rate of change of bulk energy density at nodal point *J*. In this equation, Ω denotes either the domain of a finite element, or the entire domain under consideration, and $\boldsymbol{n}$ is the outward pointing normal to the boundary surface of Ω, denoted Γ. The flux divergence term has been integrated by parts, resulting in the boundary integral, which is useful for specifying a boundary flux to the domain. Although not explicitly denoted in the foregoing expression, it is understood that the variables are numerical approximations to the exact variables appearing in the original PDE, and that each is approximated by the same finite-dimensional basis set. Isoparametric trilinear 8-node hexahedral elements are implemented for 3D simulations and bilinear 4-node quadrilaterals for 2D.

A discrete nonlinear system of time-dependent ordinary differential equations is obtained by requiring each weak residual equation be zero, $R_{\alpha,I} = 0$, *I*=1,...,*N*. The principal behind this weighted residual method is to require the vector of weight functions to be orthogonal to the residual equation, thereby enforcing each weak residual equation to be zero in a weighted average sense.

3.2 Time Integration

The time integration is achieved by using finite-difference approximations for time discretization. In this work, we consider a variable-step, predictor-corrector method, first described by Gresho, et al., (1980). Two time integration methods are implemented. A first order scheme employs a forward Euler (FE) predictor with a backward Euler (BE) corrector. A second order scheme employs an Adams-Bashforth predictor with a trapezoid rule corrector. The predictors are used to obtain an initial estimate of variables at the next time step, t_{n+1}, thereby improving the initial estimate of the solution vector for use in a Newton iteration scheme applied to the nonlinear system of equations. An added benefit of these methods is that they provide a convenient way to implement a truncation-error-based method for time-step selection. In the second-order method, the predicted variables for step *n+1* are given by the Adams-Bashforth predictor,

$$\boldsymbol{U}^{p,n+1} = \boldsymbol{U}^n + \frac{\Delta t_n}{2}\left(\left(2 + \frac{\Delta t_n}{\Delta t_{n-1}}\right)\dot{\boldsymbol{U}}^n - \frac{\Delta t_n}{\Delta t_{n-1}}\dot{\boldsymbol{U}}^{n-1}\right), \qquad (10)$$

where $\boldsymbol{U}$ denotes the vector of unknown solution variables at a grid point, the *n* superscript on $\boldsymbol{U}$ denotes the time level, the *p* superscript denotes a predictor vector, the over-dot denotes a time derivative, and the time step sizes are defined according to $\Delta t_n = t_n - t_{n-1}$, etc. The first-order predictor is the standard forward Euler formula. The variable-step corrector equation is defined as,

$$\dot{\boldsymbol{U}}^{n+1} = \frac{\eta}{\Delta t_n}(\boldsymbol{U}^{n+1} - \boldsymbol{U}^n) - (\eta - 1)\dot{\boldsymbol{U}}^n, \qquad (11)$$

where $\eta = 1$ for the first-order backward Euler method, and $\eta = 2$ for the second-order trapezoid rule. Time discretization in the FEM equations is achieved by applying the variable-step corrector for approximating the time derivatives of the capacitance terms (time derivatives of bulk densities and energy). If we denote the vector of bulk densities and energy at a grid point by $\boldsymbol{D} = (d_w, d_a, e)^T$, the grid-point time derivatives are approximated by

$$\dot{D}_j = \sum_{i=1}^{N_{DOF}} \frac{\partial D_j}{\partial U_i} \dot{U}_i, \qquad (12)$$

in terms of the grid-point "accelerations" $(\dot{\boldsymbol{U}})$ given by the corrector equation. N_{DOF} denotes the number of degrees of freedom per node point.

The predictor equations require that one or more vectors from previous time steps be available for their application in estimating variables at the next time step. In this work, the first step uses the initial condition as predictor, the second step uses the FE predictor with BE corrector. At the third step, two previous time derivative vectors are available for use in either of the user-specified predictor/corrector pairs.

The predictor/corrector scheme provides a method for estimating the local time truncation error, thereby providing a rational scheme for automatic time step control based on a user-specified truncation error tolerance. By using a norm-type analysis, the local time truncation error is estimated from the error norm between the predictor and corrector vectors. The time truncation error norm is defined by

$$\|E_t\| = \left(\frac{1}{N_{unk}} \sum_{i=1}^{N_{unk}} \left(\frac{U_i^p - U_i}{\overline{U}_i \tau_r + \tau_a}\right)^2\right)^{\frac{1}{2}}, \qquad \overline{U}_i = \frac{U_i^p + U_i}{2}, \qquad (13)$$

where N_{unk} is the global number of unknowns, and τ_r and τ_a are relative and absolute user-specified error tolerances. The next time step size is specified so that the estimated time truncation error equals the user-specified error (Gresho et al., 1980), resulting in the formula

$$\frac{\Delta t_{n+1}}{\Delta t_n} = \left(\frac{\varepsilon_t \beta}{\|E_t\|}\right)^m \tag{14}$$

where ε_t is a user-specified tolerance, $\beta = 2$ and $m = 1/2$ for the first-order method and $\beta = 3(1 + \Delta t_{n-1}/\Delta t_n)$ and $m = 1/3$ for the second-order method.

3.4 Linear and Nonlinear Solution Procedures

The task for the nonlinear solver is to find the solution vector $\boldsymbol{U}$ that minimizes the global residual vector, $\boldsymbol{R} = (\boldsymbol{R}_w, \boldsymbol{R}_a, \boldsymbol{R}_e)^T$, given by Eq. (9) once the time derivatives have been discretized by using the relations in Eqs. (11) and (12). The discretized system of nonlinear equations can be solved for the solution variables by Newton iteration (e.g., Gill et al., 1981); each step of the iteration requires the solution of the following linear system for the update vector, $\delta \boldsymbol{U}$,

$$\boldsymbol{J}(\boldsymbol{U}^q)\delta \boldsymbol{U}^{q+1} = -\boldsymbol{R}(\boldsymbol{U}^q), \tag{15}$$

where $\boldsymbol{J}$ is the Jacobian matrix,

$$J_{ij} = \frac{\partial R_i}{\partial U_j}. \tag{16}$$

The solution vector is updated at each iteration (q) according to

$$\boldsymbol{U}^{q+1} = \boldsymbol{U}^q + \delta \boldsymbol{U}^{q+1}, \tag{17}$$

until convergence is achieved; the stopping criteria is

$$\frac{1}{N_{unk}} \sum_{i=1}^{N_{unk}} \frac{|\delta U_i|}{|U_i|\varepsilon_r + \varepsilon_a} < 1. \tag{18}$$

The Jacobian can be computed efficiently via forward difference approximations, by exploiting the fact that most terms are sums of products of basis functions and grid variables. In this scheme, Eq. (16) is approximated by

$$\frac{\partial R_i}{\partial U_j} \approx \frac{R_i(U + \delta U \boldsymbol{e}_j) - R_i(U)}{\delta U}, \tag{19}$$

where $\boldsymbol{e}_j$ is a unit basis vector for the j-th unknown in the solution vector. This “inexact” Newton scheme (the term inexact in the present context refers to a numerical approximation of the Jacobian) is a convenient method of determining the Jacobian because any new transport parameter function or equation of state can be implemented without the need for the user to also program the gradient of the functions with respect to the solution vector variables. This is particularly helpful in the present class of problems where secondary variable calculations depend on which phases are present at a particular node point. However, special care must be taken in computing the forward differences to minimize finite-precision errors (see for example, Gill et al. 1981).

The Newton iteration scheme generates a linear system of equations, viz. Eq. (15), to be solved for each update vector. The systems are solved using a parallel processing preconditioned Krylov solver library called Aztec (Hutchinson et al., 1995). The library includes several parallel iterative solution methods, including the conjugate gradient method for symmetric positive definite systems and a number of related methods for nonsymmetric systems, e.g. the generalized minimum residual method (GMRES). The library includes several preconditioners (e.g., Jacobi, least-squares polynomial, incomplete LU decomposition), which can be “mixed and matched” with the Krylov methods. See Hutchinson et al., 1995 for additional information.

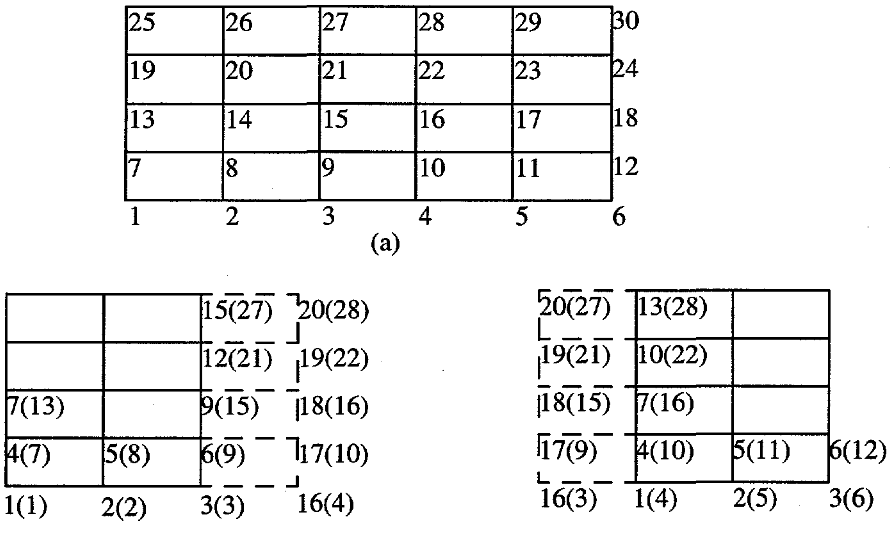
3.5 Boundary Conditions

Both Dirichlet and specified flux boundary conditions can be imposed on the discrete equations. Several combinations of Dirichlet and flux conditions are also allowable, as discussed in the following.

Fixed values for some or all degrees-of-freedom at a particular boundary node can be specified directly into the residual equation for that unknown. For the Newton iteration, the residual equation in the Jacobian matrix corresponding to the boundary node is replaced by an identity specifying the desired value. Specified surface fluxes are the “natural boundary conditions” for the FEM formulation shown earlier. Surface fluxes are conveniently applied by specifying values for the boundary flux terms appearing in Eq (9). In general the fluxes are time dependent, and can depend on the solution vector at the considered node. Note that these terms specify the net component flux, in the case of the component mass balance equations, or the total heat flux in the case of the energy equation. The fluxes can be specified in terms of transfer coefficients in the form,

$$\boldsymbol{q} \bullet \boldsymbol{n} = h(T - T_{ref}),$$

for the case of energy transport, and in analogous forms for the mass balance equations. If the parameters are dependent on the solution variables in a general way, a user-supplied function can be included to allow such an application.



(b) Subdomain for processor P0

(c)Subdomain for processor P1

Figure 1. Schematic of a domain decomposition in which the global mesh (a) is decomposed into two subdomains, (b) and (c). The dashed elements denote the "ghost elements" on each partition. The global mesh displays the global node numbering scheme, whereas the subdomains display the processor-level node numbering, with the corresponding global node numbers displayed in parenthesis.

Combinations of surface fluxes and Dirichlet conditions can also be applied, so long as they form a mathematically consistent set of boundary conditions.

3.6 Parallel Implementation

The foregoing numerical algorithm is implemented for distributed memory parallel computers, or networked systems, via domain decomposition and message-passing techniques (e.g., Foster, 1995). We used the MPSalsa code (Shadid et al., 1996) as a platform, including the parallel-processing implementation. However, there are several properties of the current problem which necessitated some special developments. A special data structure was necessary for the porous medium calculations. In our formulation it is necessary to have a unique material type specified on a node-point basis. This requirement can be attributed to the capillary pressure vs. saturation constitutive model, which is non-unique at a material boundary, and our choice of primary variables. Our solution to accommodate this non-FEM data structure was to build a processor node-point-to-material mapping. To ensure a consistent mapping across processors, inter-processor communication is required to update the correct mapping for the "ghost nodes" on each processor.

Domain decomposition, as used in the current context, refers to the process of breaking the geometrical computational domain into an assemblage of sudomains for the purpose of assigning the computational work for each subdomain to one unique processor. The domain decomposition itself is performed with a modified version of the Chaco (Hendrickson and Leland, 1993) graph partitioning code. The task for Chaco is to decompose the global node set graph into a user-specified number of partitions (subgraphs) in such a way as to minimize the edge interactions between partitions.

A simplified schematic of a domain decomposition is shown in Figure 1 for a structured grid (the algorithm is implemented for general unstructured grids). The global domain is decomposed into two subdomains, wherein the set of nodes assigned to each processor are numbered sequentially, starting with the set of nodes that "belongs" to the processor, and with the ghost or external nodes numbered last. In the figure, the global node numbers corresponding to the processor-level node numbers on the mesh partitions (Figures 1b and 1c) are shown in parenthesis. The dashed elements are the so-called "ghost elements" and the associated node points are referred to as "ghost nodes" or external nodes. Each processor is assigned the nodes corresponding to the un-dashed elements in Figure 1. However, on each processor the data for the external nodes is necessary to complete the processing for the border nodes, i.e. the surface nodes on each subdomain. Clearly, interprocessor communication is required for exchanging information associated with the ghost nodes, and those "border" nodes associated with the current processor.

The implementation of this communication requires an inter-processor communication mapping, i.e., for each processor a list of other processors with which data must be shared, and, for each of these processors, a list of nodes with which data is to be

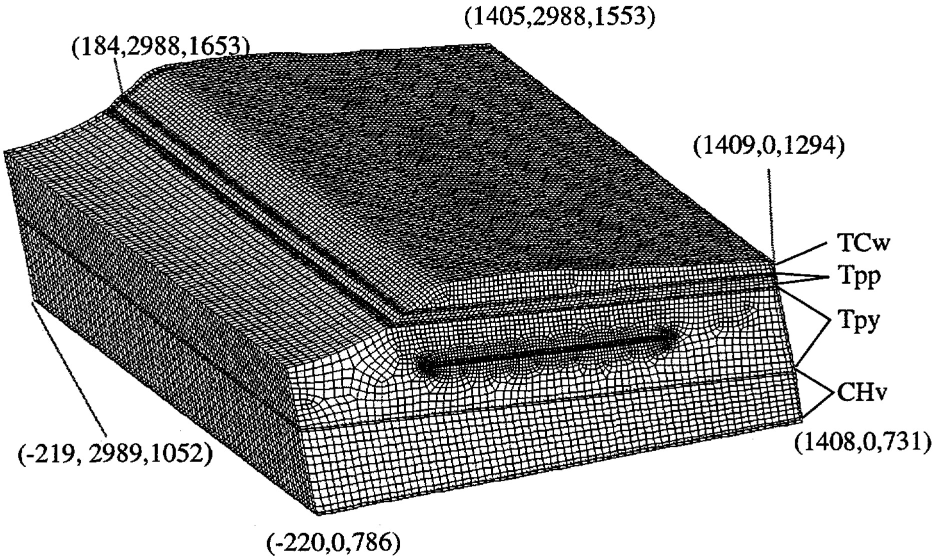


Figure 2. Grid composed of 358,000+ node points. Selected coordinates (*x,y,z*) in meters are shown for scale.

exchanged. The interprocessor communication is set up to take advantage of the native Intel Paragon communication structures, enabling efficient communication on the Intel Paragon, and is also set up to use the standard message-passing library definition MPI (Gropp, et al., 1995), thus making the algorithm highly portable. The domain-decomposition itself is done in a preprocessing step, creating a file containing the decomposition information necessary for interprocessor communication.

The decomposition in Figure 1 suggests some features of the parallel performance associated with interprocessor communication. The amount of communication for a particular decomposition is proportional to the ratio of the number density of surface nodes to internal plus surface nodes in the partition. The relative amount of communication is analogous to the rate of heat loss of a volume, which is proportional to the surface area to volume ratio. For a user faced with solving a problem on a particular mesh, there will be a more-or-less optimal number of processors (partitions) which will minimize run-time. Too few processors will render communication costs negligible but will over-load (assuming more processors are available) the work on each processor. With too many processors, communication costs will overwhelm CPU processing. With ever-increasing demands on resolution, the latter condition is less likely than the former.

4. APPLICATION AND PERFORMANCE

Verification of the numerical implementation of the nonisothermal two-phase flow algorithm in the code was accomplished by comparison with existing numerical solutions to some standard problems involving isothermal unsaturated flow, and thermal simulations of two-phase flow. The presentation of these verification studies can be found in Martinez et al., 1997. In the following we present a sample calculation designed to demonstrate the capabilities of the simulator.

4.2 Hydrothermal Simulation

In this example we examine multiphase hydrothermal transport, due to a heat-generating source distribution, in a large scale 3D region modeled after Yucca Mountain (YM), Nevada, a site being considered for placement of a high-level nuclear waste repository. The computational grid, shown in Figure 2, is composed of over 358,000 node points and includes the major hydrostratigraphic units at YM. The waste repository is to be sited in the Topopah Springs unit and is indicated by the fine discretization in the figure, where the heat source is located. This hexahedral unstructured finite element mesh was created with the CUBIT (Blacker et al., 1994) mesh generation program. This domain was meant to be similar to YM, but it is not a true representation. The front-facing cross section shown in Figure 2 is in fact representative of YM. However, the full 3D region was generated by "sweeping" this cross section backward for 3 km at a vertical inclination of about 5 degrees. Figure 2 includes spatial coordinates of selected points for scale. The entire region measures about 1.6 km east to west, includes about 0.7 km above the water table (the lower boundary) and extends 3 km along its length.

The materials are modeled as composite fractured media using the so-called equivalent continuum model (ECM) model (Peters and Klavetter, 1988) which assumes the fracture and matrix systems are locally in pressure equilibrium. The resulting models for bulk porosity, saturation, and permeability, are given by

$$\phi_b = \phi_f + (1 - \phi_f)\phi_m, \tag{20}$$

$$\phi_b S_b = S_f \phi_f + S_m (1 - \phi_f)\phi_m, \tag{21}$$

and

$$k_b = k_f \phi_f + k_m (1 - \phi_f), \tag{22}$$

where ϕ, S, and k denote porosity, liquid saturation and permeability (product of intrinsic and relative permeability), and subscripts *f*, *m*, and *b* denote fracture, matrix and bulk values. Each of the matrix and fracture capillary pressure functions, $S_m(P_c)$ and $S_f(P_c)$, is described by the van Genuchten (1978) model,

$$P_c(\bar{S}) = \frac{\rho_l g}{\alpha}\left[\bar{S}^{-\frac{1}{\lambda}} - 1\right]^{\frac{1}{\beta}}, \qquad \bar{S} = \frac{S - S_r}{1 - S_r}, \tag{23}$$

where $\lambda = 1 - 1/\beta$, and g is the gravitational acceleration. The scaled saturation, $\bar{S}$, is a function of the liquid saturation, S, and the material residual saturation, S_r. The two material parameters, α and β, influence the degree of nonlinearity in the problem. The former provides a pressure scale for capillary forces, while the latter is indicative of pore-size distribution. Individual material parameters (α_γ, β_γ, $S_{r,\gamma}$, γ=*f*, or *m*) are prescribed for the fracture and matrix functions, which, when substituted into Eq. (21), results in a nonlinear expression between the capillary pressure and bulk saturation. Similarly, the relative permeability functions for the liquid phase in the fracture and matrix is specified by a model corresponding to the van Genuchten function, developed as suggested by Mualem (1976),

$$k_{r,\gamma}(\bar{S}) = \sqrt{\bar{S}}\left[1 - \left(1 - \bar{S}^{\frac{1}{\lambda}}\right)^{\lambda}\right]^2 \tag{24}$$

in which $\bar{S}$ corresponds to fracture or matrix scaled saturation and $k_\gamma = k_{s,\gamma} k_{r,\gamma}$, γ=*f*, or *m*, and $k_{s,\gamma}$ denotes the saturated permeability for the fracture or matrix. For the gas phase, the relative permeabilities are given by the functional form $1 - k_{r,\gamma}$.

Because the saturation functions so described are nonlinear implicit functions of the capillary pressure, the functions were specified in tabular form. The FORTRAN code for these models was provided by R. R Eaton of SNL. The property data for this problem are those specified in the study of Altman et al. 1996, and are shown in Table 1. These materials, and especially with the ECM, display many orders of magnitude variation in permeability from unit to unit, rendering a highly nonlinear problem which requires the Newton iteration scheme for convergence.

Table 1. Material properties.

Unit	k (m^2)	$\alpha\times10^6$ (1/Pa)	β	ϕ^a	S_r
matrix					
TCw	3.06x 10^{-16}	7.56	1.95	0.22	0.14
tpp	7.44x 10^{-14}	34.1	1.43	0.5	0.1
tpy	4.7x 10^{-18}	1.15	1.3	0.08	0.13
CHv	1.03 x 10^{-14}	98.	1.29	0.34	0.06
Repo	4.7x 10^{-18}	1.15	1.3	0.08	0.13
fracture					
TCw	1.47x 10^{-12}	1030	3.0	7.55	10^{-4}
tpp	3.71x 10^{-13}	1025	3.0	1.91	10^{-4}
tpy	2.07x 10^{-12}	1018	3.0	12.6	10^{-4}
CHv	1.01 x 10^{-12}	1330	3.0	3.59	10^{-4}
Repo	2.07x 10^{-12}	1018	3.0	12.6	10^{-4}

a. matrix porosity ϕ_m, or fracture porosity, $\phi_f \times 10^4$

Lateral boundaries are specified as no-flow for all mass and energy balance equations. The entire lower boundary is modeled as a water table (moisture saturation unity) at 1 atmosphere pressure and a temperature of 20° C. The upper boundary is also at 1 atmosphere pressure, but at a temperature of 15° C. An infiltration flux of water at 0.1 mm/yr is applied uniformly over the entire upper boundary. These conditions simulate an upper boundary in contact with the atmosphere. The heat source term for the repository was specified as

$$Q_e = Q_0 \exp\left(-\lambda t - \frac{1}{2}\left(\frac{(y - y_0)}{\sigma}\right)^2\right),$$

with $Q_0 = 0.75$ W/m^3, $\lambda = 1/63.37\,\mathrm{yr}^{-1}$, $\sigma = 494.3$ m, and $y_0 = 1500$ m. The y-coordinate is perpendicular to the plane shown in Figure 2. The repository block measures 750m by 10m in the cross section shown in Figure 2.

The simulation was performed in two steps. In order to start with a pre-waste-emplacement condition, a steady state solution was first obtained with no repository heating. This solution pro-

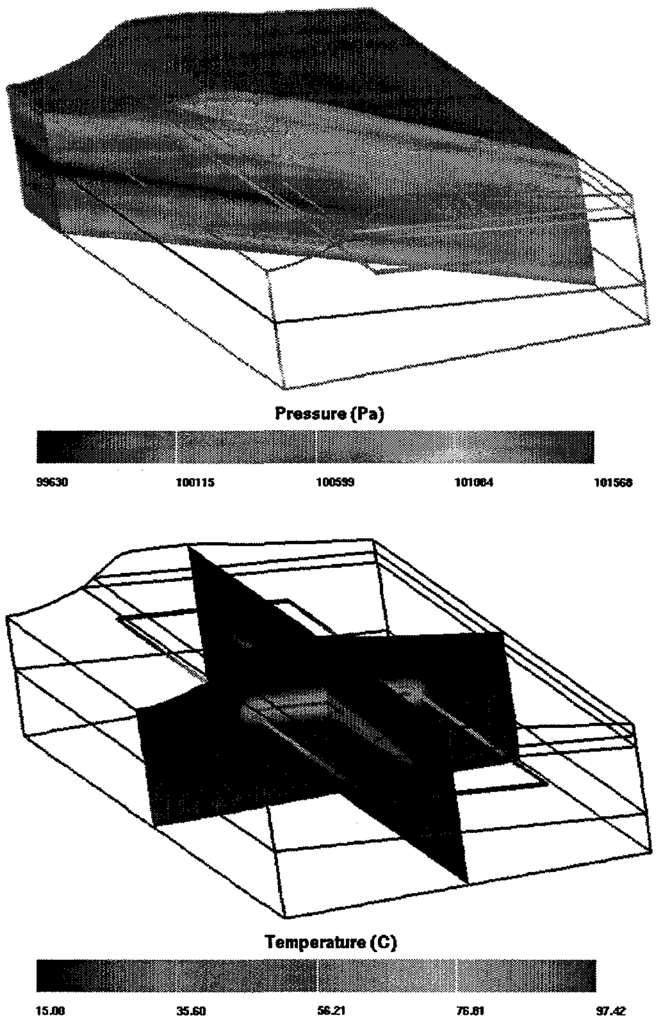


Figure 3. Pressure and temperature distributions at 23 years. Each geologic unit is modeled as fractured medium, which provides enough gas permeability to negate significant gas pressure build-up.

vided the initial condition to the next phase of the study which included repository heating.

Figure 3 shows pressure and temperature distributions after 23 years of heat output by the model repository. The temperature is very near to the (thermodynamically) saturated temperature for this pressure. Much of the air has been driven from the repository at this time. This phenomenon could be important to the question of corrosion of waste containers, a process whose chemistry may be affected by the absence of air during the hot phase of the repository. Some gas pressure build-up above the repository is noted. However, due to the high gas permeability afforded by the fractured medium, the variation in gas pressure is small. An earlier 2D simulation which modeled materials as matrix-only (unfractured) and with $Q_0 = 1.0\,\mathrm{W/m^3}$ resulted in repository temperatures on the order of 160 °C and gas pressures of 0.68 MPa after 53 years of heat output. The unfractured matrix permeabilities are ultra-low at YM, on the order of 10^{-18} $\mathrm{m^2}$ for the repository horizon, which, in the absence of fracturing, can trap the evaporated gases in and around the repository resulting in very high pressures and temperatures.

This simulation, on a mesh consisting of 358,000+ grid points, requires the solution of a linear system composed of about 1.1 million equations which are solved at each time step. Transient simulations of this type require many hundreds or thousands of time steps to simulate flow over many thousands of years. This magnitude of simulation can only be performed with parallel processing computers, as in the current simulator; this is a significant capability. The simulations were run on SNL's 1800-processor Intel Paragon computer; numerical performance studies are discussed in the following section. The capacity of the simulator depends on the size (number of processors and memory per processor) of the parallel computer available. The present example did not utilize the full capacity of the Paragon. SNL has recently procured an Intel-built 9000 processor teraflop machine. This machine enables fully transient simulations with 10^8 grid point resolution capacity.

4.2 Parallel Performance

The foregoing problem was run on various numbers of processors to assess the parallel performance on the 1800 processor SNL Intel Paragon. In order to reduce the CPU requirements for a mock heatup and cool down cycle, the heat output was reduced by setting $Q_0 = 0.5$ $\mathrm{W/m^3}$ and the simulation was carried out to 3000 years, starting with the pre-emplacement solution, as described earlier. The GMRES solver, with incomplete iLU preconditioning and row-sum scaling, was used. The iLU preconditioning algorithm was not overlapped over the processors (overlapping is available, but the memory and communication costs are substantial), hence, the number of linear solver iterations required will vary with the number of processors.

The total CPU time, excluding loading mesh data and file handling, is shown in Figure 4 as a function of the number of processors. Each simulation, performed with different numbers of processors, required 71 time steps and 125 Newton iterations to integrate the solution out to 3000 yrs. This demonstrates the correct parallel implementation of the solution algorithm. As noted above, the iLU preconditioner was not overlapped on the various processors, and therefore the number of linear solver iterations varies with the number of processors. For the present simulation, 9023, 9841, and 10103 total linear iterations were necessary on 256, 500 and 1024 processors, respectively. The speedup ratio shown is defined as

$$\frac{256 \bullet T_{256}}{T_p}$$

where T_p denotes the CPU time on p processors. The numerator estimates the CPU time on a single processor based on the CPU time for 256 processors. This estimate is not expected to be accurate (relative to the speedup ratio based on execution time on one processor), and neither is the speedup ratio for 256 processors shown on Figure 4. However, the current problem is far too large to fit on a single processor of the Intel Paragon. On a fixed size grid, such as in the present case, the efficiency will always decrease as the problem is partitioned onto more processors, which increases the relative amount of communication compared to floating point operations. This is inevitable when a fixed-size mesh is partitioned onto an increasing number of processors. At 1024 processors the relative efficiency is about 79%, a good value for an unstructured grid parallel algorithm.

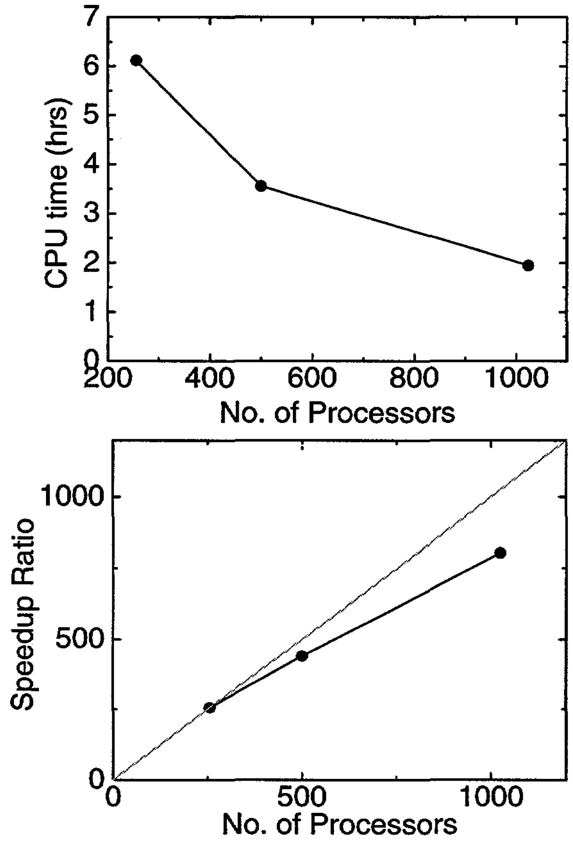


Figure 4. Parallel processing performance data for the 3D hydrothermal simulation. Each simulation required 71 time steps, and 125 Newton iterations, demonstrating a fully parallel solver implementation.

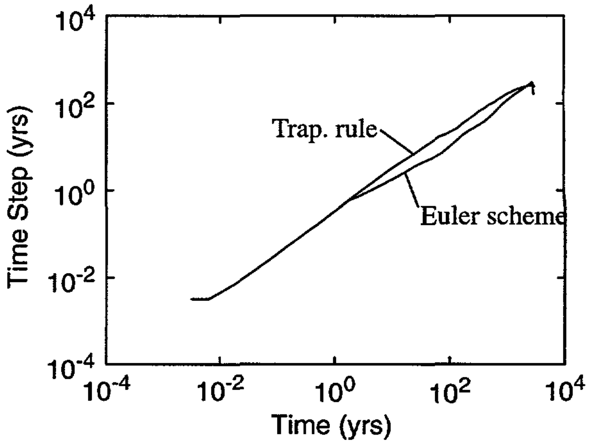


Figure 5. Time step history comparison between the first-order and second order time integration methods.

Figure 5 compares the time-step history between the second-order Adams-Bashforth/trapezoid-rule (ABT) integrator and the first-order forward-Euler/backward-Euler (FEBE) scheme. The former scheme performs the integration to 3000 years in 48 time steps (77 Newton iterations) versus 71 steps (125 Newton iterations) for the Euler scheme. On this problem, the second-order method performs the integration in about 59% of the time required by the first order method. Note that although the ABT method increased the time step sizes quicker, the time step size for the FEBE method nonetheless "caught-up" at the end of the simulation.

5. CONCLUDING REMARKS

This paper discusses the development of a 3D unstructured grid, two-phase, two-component subsurface transport simulator implemented for distributed memory parallel computers. This simulator has been run on the Intel Paragon, the IBM SP-2, as well as on a network of workstations. Through the implementation of the MPI message-passing protocols, the code can be used on any system which has MPI installed. The performance results clearly show the potential for this general-purpose MP capability. In particular, the sample application presented demonstrates that 3D, large-scale multiphase simulations with high resolution of geologic strata are feasible, and moreover can now be performed routinely on the SNL Intel Paragon. However, the parallel processing capabilities can also be accessed on (ubiquitous) networked systems via MPI. This enables 3D high-resolution parallel processing simulations to be performed on common networked systems. The numerical platform provides an excellent base for performing research in subsurface transport related activities.

ACKNOWLEDGEMENTS

This work was performed under the LDRD program at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-ACO4-94AL85000.

REFERENCES

Altman, S. J., S. Mishra, C. Li, J. T. Birkholzer, J. E. Houseworth, C. F. Tsang, S. A. McKenna, Y. W. Tsang, R. R. Eaton, V. C. Tidwell, J. S. Y. Wang, B. S. RamaRao, M. J. Martinez, P. B. Davies, 1996, Multi-scale modeling to evaluate scaling issues, percolation flux and other processes for PA recommendations, Level E Milestone T6540, Yucca Mountain Project, Las Vegas, NV, September, 1996.

Bear, J., 1972, *Dynamics of Fluids in Porous Media*, American Elsevier, New York, 764 pp.

Blacker, T. D., W. J. Bohnhoff, T. L. Edwards, J. R. Hipp, R. R. Lober, S. A Mitchell, G. D. Sjaardema, T. J. Tautges, T. L. Wilson, W. R. Oakes, S. Benzley, J. C. Clements, L. Lopez-Buriek, S. Parker, M. Whitely, D. White and E. Trimble, 1994, Sandia National Laboratories Tech. Rept., *SAND94-1100*, Sandia National Laboratories, Albuquerque, NM, 180 pp.

Foster, I., 1995, *Designing and Building Parallel Programs*, Addison-Wesley Publishing Company, Menlo Park, CA, 390 pp.

Gill, P. E., W. Murray, and M. H. Wright, 1981, *Practical Optimization*, Academic Press, New York, 401 pp.

Gresho, P. M., R. L., Lee, and R. L. Sani, 1980, On the time-dependent solution of the incompressible Navier-Stokes equations in two and three-dimensions, *Recent Advances in Numerical Methods in Fluids, Volume 1*, Pineridge Press Ltd., Swansea, U. K., 27-81.

Gropp, W., E. Lusk, and A. Skjellum, 1995, *Using MPI*, MIT Press, Cambridge, 307 pp.

Hendrickson, B., and R Leland, 1993, The Chaco user's guide, *SAND93-2339*, Sandia National Laboratories, Albuquerque, NM, 22 pp.

Huebner, K.H., and E. A. Thornton, 1982, *The Finite Element Method for Engineers*, John Wiley & Sons, New York, 623 pp.

Hughes, T. J. R., 1987, *The Finite Element Method*, Prentice-Hall, inc., Englewood Cliffs, NJ, 803 pp.

Hutchinson, S. A., J. N. Shadid and R. S. Tuminaro, 1995, Aztec user's guide, Version 1.0, Sandia National Laboratories Tech. Rept., *SAND95-1559*, Sandia National Laboratories, Albuquerque, NM, 40 pp.

Martinez, M. J., and K. S. Udell, 1990, Axisymmetric creeping motion of drops through circular tubes, *J. Fluid Mech.*, **210**, 565-591.

Martinez, M. J., 1995, Mathematical and numerical formulation of nonisothermal multicomponent three-phase flow in porous media, *SAND95-1247*, Sandia National Laboratories, Albuquerque, NM.

Martinez, M. J., P. L. Hopkins, and J. N. Shadid, 1997, LDRD Final Report: Physical simulation of nonisothermal multiphase multicomponent flow in porous media, *SAND97-1766*, Sandia National Laboratories, Albuquerque, NM, 65 pp.

Mualem, Y., 1976, A new model for predicting the hydraulic conductivity of unsaturated porous materials, *Water Resources Research*, **12** (3), 513-522.

Peaceman, D. W., 1977, *Fundamentals of Numerical Reservoir Simulation*, Elsevier, New York.

Peters, R. R., and E. A. Klavetter, 1988, A continuum model for water movement in an unsaturated fractured rock mass, *Water Resources Research*, **24**, No. 3, pp. 416-430.

Pruess, K., 1987, TOUGH user's guide, *LBL-20700*, Lawrence Berkeley Laboratory, Berkeley, CA.

Shadid, J. N., H. K. Moffat, S. A. Hutchinson, G. L. Hennigan, K. D. Devine, and A. G. Salinger, 1996, MPSALSA, A finite element computer program for reacting flow problems, Part 1 - Theoretical development, Sandia National Laboratories Tech. Rept., *SAND95-2752*, Sandia National Laboratories, Albuquerque, NM, 81 pp.

van Genuchten, R., 1978, Calculating the Unsaturated Hydraulic Conductivity with a New Closed Form Analytical Model, *Water Resources Bulletin*, Princeton University Press, Princeton University, Princeton, NJ.

Vargaftik, N. B., 1975, *Tables of the Thermophysical Properties of Liquids and Gases*, 2nd Ed., John Wiley and Sons, New York.