Advances in Water Resources 24 (2001) 763–774

Advances in
Water Resources

www.elsevier.com/locate/advwatres

Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems ☆

Jim E. Jones, Carol S. Woodward *

Lawrence Livermore National Laboratory, Center for Applied Scientific Computing, P.O. Box 808, L-561, Livermore, CA 94551, USA

Received 26 April 2000; received in revised form 13 November 2000; accepted 14 December 2000

Abstract

In this paper, we present a class of solvers developed for the parallel solution of Richards' equation, a model used in variably saturated flow simulations. These solvers take advantage of the fast, robust convergence of globalized Newton methods as well as the parallel scalability of multigrid preconditioners. We compare two multigrid methods. The methods differ primarily in their handling of discontinuous and anisotropic permeability fields, with one method invoking a simple pointwise smoothing technique and the other a more expensive plane smoother. Computational results are presented to show the effectiveness of the entire nonlinear solution procedure, to demonstrate the effect of discontinuities and anisotropies, and to explore parallel efficiencies. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Variably saturated flow; Richards' equation; Preconditioning; Newton–Krylov; Multigrid; Anisotropy

1. Introduction

In recent years we have observed an increase in attention to modeling the flow of water through variably saturated porous media. This increase arises from heightened interest in managing limited water resources, in finding appropriate sites for waste facilities and in evaluating the impact of current sites on local groundwater systems. One way to gain an understanding of the groundwater systems at these sites is through computer simulations of subsurface flow. However, understanding water movement in these sites usually requires full regional models. These models are challenging to solve numerically, as they have extremely large numbers of spatial zones (usually in the millions), and they exhibit nonlinearities and large variations in equation coefficients.

The nonlinearities generally necessitate some sort of iteration for solution of the discrete equations at each time-step. These iterations require the solution of large, nonsymmetric linear systems. In addition, the variation in equation coefficients makes these linear systems difficult for many traditional methods to solve. Typically, the large variation in coefficients is due to the use of a geostatistical model for permeability, allowing many orders of magnitude change in the permeability from one cell to the next (heterogeneity) as well as correlations of values in each direction (statistical anisotropy). High heterogeneity and anisotropy in the problem coefficients make the problem difficult to solve numerically.

In addition, the timely solution of high-resolution discretizations of these region-scale problems requires scalable algorithms. An algorithm is scalable if the computational complexity is in the order of the number of unknowns, $O(N)$, and not some power of this number. For example, Gaussian elimination requires $O(N^3)$ operations, so doubling the number of unknowns multiplies the computational work by eight. This algorithm is not scalable because the computational work increases exponentially with problem size. Some multigrid methods, however, have computational complexities of $O(N)$. So, doubling the number of unknowns only doubles the computational work. If this algorithm is implemented well on a parallel computer, we might expect that doubling the number of unknowns and the number of processors leaves us with a fairly constant

[☆] This work was performed under the auspices of the US Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

[*] Corresponding author.

E-mail addresses: jjones@llnl.gov (J.E. Jones), cswoodward@llnl.gov (C.S. Woodward).

0309-1708/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved.
PII: S0309-1708(00)00075-0

computation time. Thus, we pursue scalable algorithms with the goal of effectively using parallel computation to solve large-scale variably saturated flow problems.

Many authors have considered solution methods for models of variably saturated flow [8,15,18,25,19,21,28]. In particular, Celia, Bouloutas and Zarba developed the modified Picard method which solves the nonlinearities in the problem while maintaining global conservation of mass using a finite element method [8]. However, this method does not exhibit local mass conservation. Tocci, Kelly and Miller proposed solution of the Richards' equation model using the method of lines approach for time discretization, finite-differences for the spatial discretization and Newton's method for solving the nonlinearities at each time-step [25]. They, together with Williams, addressed some subtleties of using the Newton's method approach with interpolated approximations to saturation and relative permeabilities in [19]. They did not, however, address the issues of highly resolved, large-scale domains. Ross and Bristow [21] considered using the Kirchhoff transformation in order to better approximate the solution in cases where the absolute permeability is a discontinuous function of space. In [28] Williams, Miller and Kelley extended this work to a family of transformation methods.

With either a finite-difference or finite element discretization method, a discrete system of coupled, nonlinear equations must be solved at each time-step. The solvers to be used must handle highly heterogeneous and anisotropic media, as well as problems with large number of unknowns. In this paper, we describe a family of solution methods for the discrete equations arising at each time-step in variably saturated flow problems. These solvers use Newton's method for the solution of the implicit, nonlinear equations and a Krylov iterative solver to solve the Jacobian systems. The iterative, linear solver is preconditioned with a semicoarsening multigrid algorithm. By combining the nonlinear Newton iteration with a multigrid preconditioner, we hope to take advantage of the fast, robust nonlinear convergence of Newton's method and the scalability of the linear multigrid method.

Multigrid methods have been shown to be effective solvers for the discrete equations arising from discretized elliptic partial differential equations. Multigrid solvers are iterative solvers, and their chief advantage is that they are *algorithmically scalable*, i.e., the convergence rate is independent of the size of the discretized system. In groundwater applications, the multigrid solver must be able to deal with jumps in the permeability coefficient and anisotropies induced either by the problem's coefficients or by the grid. Early work by Dendy and colleagues [1,9] showed effective multigrid solvers for such problems required modifying the interpolation to respect jumps in the diffusion coefficient. This work also demonstrated the ability of alternating line relaxation to deal with fairly general anisotropies. Later work [11] showed the effectiveness of multigrid based on semicoarsening and line relaxation. This work also showed multigrid to be competitive with ILU solvers on small 3D reservoir simulation problems.

In this paper, we compare the effectiveness of two multigrid methods (ParFlow multigrid solver (PFMG) [2] and semicoarsening solver (SMG) [23]) used as preconditioners within a Newton–Krylov solver for Richards' equation. For large saturated groundwater problems, the PFMG was shown to be several orders of magnitude faster than the diagonally scaled conjugate gradient solver [2]. In the context of radiation diffusion, the SMG solver was shown to be significantly faster than the diagonally scaled conjugate gradient solver and generally faster than incomplete factorization methods, particularly for large problems [3]. As we are interested in large-scale simulations on parallel computers, the algorithmic scalability of multigrid methods and the existence of efficient parallel implementations [2,6] makes them an attractive alternative to more standard solvers. The focus of this paper is on how the performance of these two multigrid methods depends on the statistical characteristics of the permeability field.

The rest of this paper is organized as follows. In Section 2, we present the formulation of Richards' equation that we use and the discretization method we employ. In Section 3, we overview the nonlinear Newton and linear Krylov iterative methods used for the solution of the implicit system at each time-step. Section 4 discusses the preconditioners we invoke to accelerate the Krylov method. These preconditioners include two semicoarsening multigrid schemes. We present numerical results with these two preconditioners in Section 5. Lastly, Section 6 gives some conclusions.

2. Problem formulation and discretization

We consider the mixed form of Richards' equation [20] as our model for variably saturated flow,

$$\frac{\partial(s(p)\rho\phi)}{\partial t} - \nabla\left(\frac{k(x)k_{\mathrm{r}}(p)\rho}{\mu}(\nabla p - \rho g \nabla z)\right) = q, \qquad (1)$$

where $s(p)$ is the water saturation at pressure p, ρ the water density, ϕ the porosity of the medium, $k(x)$ the absolute permeability of the medium, $k_{\mathrm{r}}(p)$ the relative permeability of water to air, μ the water viscosity, g the gravity, q represents any water source terms and z is the elevation. The equation is completed by adding boundary conditions and an initial condition as follows,

$$p = p_{\mathrm{D}} \text{ on } \Gamma^{\mathrm{D}}, \quad -\left(\frac{k(x)k_r(p)\rho}{\mu}(\nabla p - \rho g \nabla z)\right)\mathbf{n} = g_{\mathrm{N}} \text{ on } \Gamma^{\mathrm{N}} \qquad (2)$$

and

$$p = p^0 \quad \text{for } t = 0, \tag{3}$$

where Γ^{D} and Γ^{N} comprise the boundary of the problem domain, and $\mathbf{n}$ is an outward, unit, normal vector to Γ^{N}.

Note here that we use a scalar absolute permeability model. Since we are interested in highly resolved models, we are assuming that the permeability can be specified on a fine scale. In addition, we use a geostatistical model for the absolute permeability. This model allows specification of correlation lengths in each coordinate direction. As a result, we do not specify permeability as a vector or tensor field. In discussions throughout the paper, we refer to anisotropy in the problem. By this term, we mean statistical anisotropy introduced by the correlation lengths, and also physical anisotropy introduced by having significant variation in permeability value in a given direction, independent of other directions and not necessarily grid-aligned.

Discretization is done for time with an implicit backward Euler differencing scheme. Our solvers are valid for higher order time-stepping methods, but to keep this exposition simpler, we present the discrete systems arising from a first-order in time method. For spatial discretization, we use a tensor product grid with N_x, N_y and N_z cells in the x, y and z directions, respectively. We then apply cell-centered finite-differences over this mesh. We use harmonic averaging for interface values of the absolute permeability and one-point upstream weighting for interface values of relative permeability. These discretization methods result in a coupled system of nonlinear equations that must be solved at each time-step, where the equation at each cell is given by,

$$\begin{aligned} F_{i,j,k}(p^n) = {} & \Delta x_i \Delta y_j \Delta z_k \phi_{i,j,k} \rho \left(s(p)^n_{i,j,k} - s(p)^{n-1}_{i,j,k} \right) \\ & - \Delta t^n \Delta x_i \Delta y_j \Delta z_k q^n_{i,j,k} - \Delta t^n \Delta x_i \Delta y_j \Delta z_k \\ & \times \left(\frac{U^x_{i+1/2,j,k} - U^x_{i-1/2,j,k}}{\Delta x_i} + \frac{U^y_{i,j+1/2,k} - U^y_{i,j-1/2,k}}{\Delta y_j} \right. \\ & \left. + \frac{U^z_{i,j,k+1/2} - U^z_{i,j,k-1/2}}{\Delta z_k} \right) = 0, \end{aligned} \tag{4}$$

where

$$U^x_{i+1/2,j,k} \equiv \left(\frac{k(x) k_{\mathrm{r}}(p^n) \rho}{\mu} \right)_{i+1/2,j,k} \frac{p^n_{i+1,j,k} - p^n_{i,j,k}}{\Delta x_{i+1/2}} \tag{5}$$

is the $i + 1/2, j, k$ interface value of the flux x-component. Other components are defined similarly.

We apply this spatial discretization method because it is locally conservative, and, in the linear case, is equivalent to the lowest order mixed finite element method with certain quadrature [27]. Thus, this method can be shown to be second-order in space if harmonic averaging is used for $k(x)$ and $k_{\mathrm{r}}(p)$. Discretization errors for Richards' equation with a similar method have been analyzed entensively in [29]. We use one-point upstream weighting for $k_{\mathrm{r}}(p)$ to best capture fronts in the solution as indicated in [14].

It should be noted that as long as we solve the discrete nonlinear system to a tolerance below the error in the discretization method, we will not change the accuracy of the solution. In the numerical studies that follow, we choose a small enough tolerance that the solution error is dominated by discretization error and is thus unaffected by the choice of nonlinear solver.

3. Newton–Krylov methods

We employ a Newton–Krylov method to solve the nonlinear discrete problem at each time-step. This system requires the solution, p^* of $F(p) = 0$, where the i, j, kth element of the vector F evaluated at p^n is given as in (4).

Newton–Krylov methods were first used in the context of solving partial differential equations by Brown and Saad [7]. In this method, the coupled nonlinear system resulting from discretizing a partial differential equation is solved by first applying a Newton linearization, then using an iterative Krylov method to solve the resulting Jacobian systems for each Newton iteration. One big advantage to these methods is that the Krylov linear solver requires only matrix-vector products, which, since the system matrix is the Jacobian of the nonlinear function, can be approximated by taking differences of the nonlinear function,

$$J(p^k)v \approx \frac{F(p^k + \epsilon v) - F(p^k)}{\epsilon}, \tag{6}$$

where p^k is the current iterate at the kth Newton step. The value of ϵ is computed via the formula given in [7],

$$\epsilon = \operatorname{sign}(p^k v) \sqrt{uround} \ \max\{|p^k v|, \|v\|_1\} / v^{\mathrm{T}} v, \tag{7}$$

where *uround* is the machine unit round-off function.

Thus, only the implementation of the nonlinear function is necessary, and matrix entries need never be formed. A standard Newton method is locally quadratic in convergence so that once the iterate, p^k, is close enough to the solution we have [12],

$$\|p^{k+1} - p^*\| \leqslant C \|p^k - p^*\|^2. \tag{8}$$

Applying the finite-difference method in (6) gives rise to a method that is quadratic as long as ϵ is chosen small enough [5].

An inexact Newton method results from a Newton method where the linear systems are only approximately solved at each step. We use a method proposed by Eisenstat and Walker to determine the linear system tolerances [13]. In particular, at the kth step, we iterate on the linear system, $J(p^k)s^k = -F(p^k)$, until the relative residual is less than η^k where

$$\eta^k = \max\{\tilde{\eta}^k, \gamma(\eta^{k-1})^2\} \quad \text{and} \quad \tilde{\eta}^k = \gamma\left(\frac{\|F(p^k)\|}{\|F(p^{k-1})\|}\right)^2. \tag{9}$$

Here, $F(p^k)$ and $F(p^{k-1})$ are the function evaluated at the current and previous nonlinear solution iterates, respectively, and γ is 0.9. Our choice of η^k reflects the amount of decrease between the function evaluated at the current nonlinear solution iterate and the function at the previous iterate. Since the error in Newton's method decreases quickly once we are close to the solution, the residual decreases rapidly close to the solution. Thus, close to the solution, the tolerance is low, requiring precise approximate solves of the Jacobian system. Further away, the tolerance is high, preventing "over-solves" of the linear system when a highly precise approximation to the Jacobian system solution will not provide much more value than a coarse approximation to the nonlinear system solution. Under suitable assumptions, the local convergence of this method can be shown to be q-quadratic [13].

To enhance the robustness of the Newton method, we added the line-search backtracking procedure detailed by Brown and Saad [7]. This procedure allows adjustment of the Newton step to guarantee progress towards the solution of the nonlinear problem at each iteration. The step taken obeys the Goldstein–Armijo conditions [12] ensuring that we have sufficient decrease in the nonlinear function relative to the step length, that we obey a minimum step length relative to the rate of decrease at the current iterate, and that we take the full Newton step close to the solution. In [19], it is pointed out that for certain regimes, the van Genuchten curves have derivatives that are not Lipschitz continuous. Under these conditions, standard Newton methods can fail. The authors of [19] point out that using integrated interface permeabilities with Hermite splines can allow convergence in these regimes while maintaining efficiency. Transformation methods applied to Richards' equation can also contribute to robustness for these situations [28]. In preliminary experiments, we have found that line-search may also allow convergence in some of these difficult regimes, and this is the technique we have used in this work.

Furthermore, the line-search procedure allows convergence of the method even when the initial guess is not local to the solution. The method, in a sense, can bring the iterates into the radius of convergence for the Newton method. In the case of Richards' equation, we take the approximate solution from the previous time-step as the initial guess for the nonlinear solution at the new time-step. When the time-step is large, this initial guess may not be close to the solution for the new time. However, line-searches allow larger steps than may be possible with a local method.

As water resource problems can be very large, we apply a preconditioned iterative process for the Jacobian system solution. Furthermore, the Jacobian of a nonlinear elliptic operator is nonsymmetric, so we use the Krylov method GMRES [22]. This method was developed for nonsymmetric linear systems and has the advantages that in exact arithmetic, the linear system residual is a nonincreasing function of iteration, and convergence is guaranteed after n iterations for a linear system of size $n \times n$.

This method builds up a Krylov basis with the matrix and initial residual, adding one vector to this basis with each iteration. These basis vectors are then needed in future iterations to generate new iterates. As a result of building this basis, the memory requirements for the method increase with the number of iterations. One way to reduce these requirements is to use the restarted version of the method, which builds the Krylov basis up to m vectors, then restarts the method using the mth iterate as the initial guess. By restarting, the memory requirements for GMRES are reduced significantly as only m Krylov basis vectors need to be saved at any time. The disadvantage of the restarted method is that convergence is no longer guaranteed.

Although with GMRES, the linear residual decreases with each iteration, the decrease can be quite small. To make more progress in reducing the residual at each iteration, the linear system can be preconditioned. This amounts to solving the system, $M^{-1}Js = M^{-1}(-F)$, where M is an approximation to the Jacobian, and systems like $Mr = b$ are easy to solve. In this case, preconditioned GMRES iterations make more progress at each step and thus will require fewer overall steps.

4. Preconditioners

In this work, we precondition the linear system with a symmetric approximation to the Jacobian. We solve the preconditioner systems, $Mr = b$, by applying one V-cycle of a multigrid method.

4.1. Preconditioning with a symmetric approximate Jacobian

To see how we form a symmetric approximation to the Jacobian we consider two structurally symmetric Jacobian matrix entries. For simplicity, we consider one-dimension, a homogeneous medium and no gravity. The component of the nonlinear function at the kth Newton iteration for time-step n corresponding to the ith cell, $F_i^{n,k}$, depends on the interface velocities $v_{i+1/2}$ and $v_{i-1/2}$, each of which depends on the relative permeability evaluated at its respective interface. These relative permeabilities are upstream weighted functions of pressure. Taking derivatives, we see that

$$\frac{\partial F_i^{n,k}}{\partial p_{i+1}^{n,k}} = C\left(\frac{\partial k_{\mathrm{r}}(p^{n,k})_{i+1/2}}{\partial p_{i+1}^{n,k}}(p_i^{n,k} - p_{i+1}^{n,k}) - k_{\mathrm{r}}(p^{n,k})_{i+1/2}\right) \tag{10}$$

and

$$\frac{\partial F_{i+1}^{n,k}}{\partial p_i^{n,k}} = C\left(-\frac{\partial k_{\mathrm{r}}(p^{n,k})_{i+1/2}}{\partial p_i^{n,k}}(p_i^{n,k} - p_{i+1}^{n,k}) - k_{\mathrm{r}}(p^{n,k})_{i+1/2}\right), \tag{11}$$

where C is a constant depending on the mesh spacings and physical constants of the problem. Thus, the Jacobian is nonsymmetric because of the sign on the relative permeability derivative and also because the relative permeability derivative is taken with respect to a different pressure in each of the above equations. In this work, we form an approximation by simply dropping the first right-hand side terms in (10) and (11). This approximation degrades as an approximation to the true Jacobian in areas where the relative permeability changes rapidly with pressure. For our current purposes, this approximation provides an effective preconditioner. Other approximations are possible. See [17] for further discussion.

4.2. *Multigrid preconditioners*

Multigrid methods can be very efficient solvers for the linear systems arising from discretized elliptic partial differential equations. Multigrid's chief advantage is that it is a scalable algorithm: when properly designed, the solver's convergence rate is independent of the size of the discretized system [4]. Standard multigrid methods combine simple relaxation (which quickly reduces high-frequency error components) with error correction from a coarser grid (which can accurately represent low-frequency error components). For our problem, the multigrid solver must be able to efficiently deal with anisotropies and widely variable coefficients. In the numerical results that follow, we compare the performance of two multigrid algorithms within the context of our nonlinear solution procedure: the PFMG developed by Ashby and Falgout [2], and the SMG developed by Schaffer [23] (see also [10] and [24]).

Let $AU = F$ be the given linear system to solve. The unknown U and right-hand side F are vectors defined on a logically rectangular grid. We will use an h superscript to denote quantities defined on the given grid. The matrix A is symmetric, positive definite, and connections have the standard "nearest-neighbor" 7-point stencil form. As the grid is logically rectangular, there is a unique index (i,j,k) for each point on the grid. One can produce a grid that is coarser in, say, the z-direction, by considering only the points $\{(i,j,k),\ k \text{ odd}\}$. We will use a $2h$ superscript to denote quantities defined on the coarse grid. This procedure is called semicoarsening (as opposed to full or standard coarsening) as the coarse grid is only coarser in one of the dimensions. Coarsening only in the direction of strong coupling (as opposed to all directions) results in a multigrid algorithm that is effective for anisotropic problems. The multiple grids are used in a multigrid V-cycle as outlined below [4].

```
$V(v_1, v_2)$-cycle
  (1) Pre-relaxation on $A^h U^h = F^h$. Perform $v_1$ sweeps of relaxation.
  (2) Set $F^{2h} = I_h^{2h}(F^h - A^h U^h)$.
  (3) "Solve" $A^{2h}U^{2h} = F^{2h}$ by recursion.
  (4) Correct $U^h \leftarrow U^h + I_{2h}^h U^{2h}$.
  (5) Post-relaxation on $A^h U^h = F^h$. Perform $v_2$ sweeps of relaxation.
```

A multigrid method is determined by several components: the relaxation method (typically a simple iterative method like Gauss–Seidel), the interpolation operator I_{2h}^h used to transfer vectors from coarse to fine grids, the restriction operator I_{2h}^h used to transfer vectors from fine to coarse grids (in the methods we consider, restriction is the transpose of interpolation), and the coarse operator A^{2h}. Note the equation to be solved in step 3 above typically has the same form as the original grid h problem. It is solved by applying the same algorithm using a still coarser grid $4h$. Eventually, a coarse enough grid is reached and the problem can be solved directly.

In the PFMG algorithm, pointwise red/black Gauss–Seidel relaxation is used. To achieve robustness for grid-induced anisotropies, the grid is coarsened in the direction with the smallest grid spacing and thus the tightest coupling. The interpolation is based pointwise on the operator as in Dendy's Black Box multigrid [9]. Assume semicoarsening in the z-direction, and consider a point (i,j,k) not on the coarse grid. Let

$$\begin{aligned} &a_{ijk}^{\mathrm{c}} u_{i,j,k} + a_{ijk}^{\mathrm{w}} u_{i-1,j,k} + a_{ijk}^{\mathrm{e}} u_{i+1,j,k} + a_{ijk}^{\mathrm{s}} u_{i,j-1,k} \\ &\quad + a_{ijk}^{\mathrm{n}} u_{i,j+1,k} + a_{ijk}^{\mathrm{l}} u_{i,j,k-1} + a_{ijk}^{\mathrm{u}} u_{i,j,k+1} = f_{ijk} \end{aligned} \tag{12}$$

be the equation at the point. The superscripts c, w, e, s, n, l, and u stand for central, west, east, south, north, lower, and upper, respectively. We split the operator A^h into an operator T in the coarsened direction (z) and an operator B in the other two directions as follows,

$$A^h u = T^h u + B^h u, \tag{13}$$

$$\begin{aligned} T^h u = &\left(a_{ijk}^{\mathrm{c}} + a_{ijk}^{\mathrm{w}} + a_{ijk}^{\mathrm{e}} + a_{ijk}^{\mathrm{s}} + a_{ijk}^{\mathrm{n}}\right) u_{i,j,k} \\ &+ a_{ijk}^{\mathrm{l}} u_{i,j,k-1} + a_{ijk}^{\mathrm{u}} u_{i,j,k+1}, \end{aligned} \tag{14}$$

$$B^h u = -\left(a^{\mathrm{w}}_{ijk} + a^{\mathrm{e}}_{ijk} + a^{\mathrm{s}}_{ijk} + a^{\mathrm{n}}_{ijk}\right)u_{i,j,k} + a^{\mathrm{w}}_{ijk}u_{i-1,j,k} + a^{\mathrm{e}}_{ijk}u_{i+1,j,k} + a^{\mathrm{s}}_{ijk}u_{i,j-1,k} + a^{\mathrm{n}}_{ijk}u_{i,j+1,k}. \tag{15}$$

The interpolation weights are given by a 1D operator-based interpolation using T, the operator in the coarsened direction. In particular, the point (i,j,k) interpolates from the coarse grid points directly above and below it (indices $(i,j,k+1)$ and $(i,j,k-1)$ on the fine grid) with weights,

$$\begin{aligned} w^{\mathrm{u}}_{ijk} &= -a^{\mathrm{u}}_{ijk}/(a^{\mathrm{c}}_{ijk} + a^{\mathrm{w}}_{ijk} + a^{\mathrm{e}}_{ijk} + a^{\mathrm{s}}_{ijk} + a^{\mathrm{n}}_{ijk}), \\ w^{\mathrm{l}}_{ijk} &= -a^{\mathrm{l}}_{ijk}/(a^{\mathrm{c}}_{ijk} + a^{\mathrm{w}}_{ijk} + a^{\mathrm{e}}_{ijk} + a^{\mathrm{s}}_{ijk} + a^{\mathrm{n}}_{ijk}), \end{aligned} \tag{16}$$

respectively. For fine grid points also on the coarse grid, interpolation is the identity; the interpolated fine grid value is equal to the coarse grid value. Unlike linear interpolation, the interpolation weights given in (16) are affected by variations in the operator A^h and result in a method that is robust for widely variable coefficients.

The coarse grid operator in the coarsened direction is formed by the Galerkin conditions $T^{2h} = I^{2h}_h T^h I^h_{2h}$. In the other two directions, a weighted sum of coefficients is used. In particular, consider a fine point (i,j,k) also on the coarse grid. The south coefficient in the coarse grid operator B^{2h} is given by

$$a^{2h,\mathrm{s}}_{i,j,k} = a^{h,\mathrm{s}}_{i,j,k} + \frac{1}{2}\left(a^{h,\mathrm{s}}_{i,j,k+1} + a^{h,\mathrm{s}}_{i,j,k-1}\right). \tag{17}$$

Similar averages define the other off-diagonal elements of B^{2h}. As on the fine grid, the diagonal element of B^{2h} is defined as minus the sum of the off-diagonals. The coarse grid operator is then defined by $A^{2h} = T^{2h} + B^{2h}$ resulting in a 7-point stencil for A^{2h}.

In the SMG algorithm, plane-wise red/black Gauss–Seidel relaxation is used. The values of the current approximation on the odd planes are updated simultaneously to satisfy the equations on these planes. This is followed by a similar update on the even planes. These plane solves are approximated by one 2D multigrid V-cycle. This relaxation provides robustness for anisotropic problems where the strong coupling is within the planes. Semicoarsening provides robustness for strong coupling orthogonal to the planes.

An important, unique feature of the SMG algorithm is the definition of the interpolation operator I^h_{2h}. The definition is motivated by the relationship between error on odd and even planes after an even plane relaxation sweep. To briefly motivate the approach, consider solving the tridiagonal system $Au = b$.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot\quad\cdot & \cdot \\ & & & & a_{n,n-1} & a_{n,n} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ f_n \end{bmatrix}. \tag{18}$$

After relaxing the even numbered equations, the approximate solution u satisfies equations

$$a_{k,k-1}u_{k-1} + a_{k,k}u_k + a_{k,k+1}u_{k+1} = f_k, \quad k = 2,4,6,\ldots \tag{19}$$

Since the exact solution $\bar{u}$ satisfies the same equations, the error $e = \bar{u} - u$ satisfies the equations

$$a_{k,k-1}e_{k-1} + a_{k,k}e_k + a_{k,k+1}e_{k+1} = 0, \quad k = 2,4,6,\ldots \tag{20}$$

From this we can write the error at even points in terms of the error at odd points

$$e_k = -\frac{a_{k,k-1}}{a_{k,k}}e_{k-1} - \frac{a_{k,k+1}}{a_{k,k}}e_{k+1}, \quad k = 2,4,6,\ldots \tag{21}$$

We can use this relation to eliminate the even numbered variables and solve the error equation $Ae = f - Au$ only for errors at odd points. Applying this method recursively yields a cyclic reduction solver for the tridiagonal system (18). The SMG interpolation is based on a generalization of this approach to block-tridiagonal systems.

Our discrete Jacobian system has a block-tridiagonal structure; grouping unknowns by z-planes, the structure is like (18) except u_k now represents all unknowns in the Kth plane. To be precise, let

$$A_{K,K-1}U_{K-1} + A_{K,K}U_K + A_{K,K+1}U_{K+1} = F_K \tag{22}$$

be the discrete equations for the Kth plane. Here U_K is a vector composed of all unknowns with z index equal to K, i.e., $U_K = (U_{i,j,K},\ i = 1,\ldots,n_x,\ j = 1,\ldots,n_y)$, where n_x and n_y are the grid sizes in the x and y directions. $U_{K\pm1}$ is defined similarly. The matrix $A_{K,K}$ contains couplings within the Kth plane; the matrix $A_{K,K-1}$ couples the Kth plane to the $K-1$th plane; and $A_{K,K+1}$ is defined similarly. After relaxing the even planes, the error on even planes can be written in terms of the error at odd planes in a way completely analogous to (21)

$$E_K = -A^{-1}_{K,K}A_{K,K-1}E_{K-1} - A^{-1}_{K,K}A_{K,K+1}E_{K+1}, \quad K = 2,4,6,\ldots \tag{23}$$

Using the odd planes as the coarse grid and the above relation to define an interpolation operator I^h_{2h} makes the multigrid V-cycle a direct method. However, this is not a practical approach in that using (23) leads to nonsparse interpolation operators. In the SMG algorithm, sparse approximations to these ideal interpolation operators are used. The matrices $-A_{K,K}^{-1}A_{K,K-1}$ and $-A_{K,K}^{-1}A_{K,K+1}$ are approximated by diagonal matrices with the same action on constant vectors. The computation of these interpolation operators involves a plane solve for each even grid plane, which we approximate with one 2D multigrid V-cycle. The coarse grid versions of A are defined by the Galerkin condition, i.e., $A^{2h} = I_h^{2h}A^hI_{2h}^h$. This results in a slight growth in the size of the coarse grid stencils. The fine grid stencil is 7-point and all coarser grid stencils are 15-point.

The PFMG method is cheaper per iteration than the SMG method in both floating point operations and storage. As we will see in the numerical results section, when both methods work well, PFMG is faster. SMG is generally more robust, however. In particular it can deal with anisotropies that vary in direction throughout the grid. This robustness is due to the use of semicoarsening *and* plane relaxation. SMG, because it faithfully follows the Galerkin condition, is provably convergent for symmetric positive definite matrices. Also, the fill-in in the coarse grid stencil allows SMG to more accurately represent anisotropies that are not grid-aligned on coarser grids.

5. Numerical results

We conducted a number of numerical tests to study how these two multigrid preconditioners performed for problems with varying amounts of heterogeneity and anisotropy in the permeability field. Often, geostatistical models are used to describe the subsurface permeability. We use the turning bands algorithm [26] with the exponential covariance model. The algorithm implementation allows specification of the mean value, standard deviation (S.D.) and directional correlation lengths. The S.D. describes how much variation, or heterogeneity, exists between uncorrelated cells. The difference in correlation lengths describes the amount of variation, or statistical anisotropy, there is from one direction to the next. Note that, as mentioned earlier, problems can also exhibit physical anisotropy that contributes to preferential flow through the domain and is modeled by either a tensoral permeability or layers not aligned with the grid. Many problems of interest are exhibiting heterogeneity and physical as well as statistical anisotropy.

We have compared the performance of the two multigrid methods described above in the context of a test case based loosely on the Lawrence Livermore National Laboratory (LLNL) site. The problem domain was 300 m × 1000 m × 120 m. The mean permeability was 1 m/day with correlation lengths of 50 m in x, 100 m in y and 5 m in z. Permeability values were cut-off below 10^{-6} and above 10^2. Porosity was 30%. Initial conditions were taken in hydrostatic equilibrium with the pressure head at the domain bottom set to 90 m. No flow boundary conditions were imposed on the left, right and bottom of the domain, with 0.1 ft/year inflow flux on the top. Dirichlet conditions were imposed on the front and back of the domain in such a way as to impose a 20 m pressure head gradient from front to back. In all cases, relative permeability and saturation curves were evaluated with van Genuchten formulas [16]. Exact formulas were used with no curve-fitting. The parameters α and n were set to 0.1/m and 2, respectively. The residual saturation was 20% and the domain was considered saturated at 99%. We used an initial time-step of 0.1 day and doubled the time-step size after every successful time-step. We ran to a final time of 25 days. If a time-step failed, we cut the step size by 1/2 and recomputed the step, growing the step size again if the recomputation was successful.

Stopping criteria for the nonlinear solver was as follows. The solver stopped if the nonlinear residual, $\|F\|$, from (4) was less than 10^{-6}. We choose this value to assure that the iteration error would be smaller than the discretization error, and thus, solutions with different solvers would be the same.

The first experiment showed the effect of changing the S.D. while maintaining the rest of the problem parameters. Table 1 shows the results of changing the variance (or square of the S.D.) from 0 to 25 for this problem. We have included this large span of variances to account for the need of modelers to study effects of uncertainties in their models. The higher variances are not expected to occur often in practice, but a useful code should have a robust solver at least as an option so that modelers can run with these extreme values if they need to understand some effect.

We see that as the variance increases, the number of nonlinear and linear iterations required to solve to the final time of 25 days increases. After a variance of 10, PFMG fails to solve three consecutively decreased time-steps, and SMG starts to show significant increases in iteration counts. However, SMG is robust enough to withstand the extreme heterogeneity posed by these problems and allow the nonlinear solver to make progress. Compute times show that PFMG gives rise to a faster solution when it is successful, but the range of heterogeneity in which it is successful is quite small.

In this data, we see that changing the preconditioner leads to changes in the nonlinear solver iteration counts. These changes are due to one of two reasons. The first reason is that an inferior preconditioner may fail to solve the linear problem to the specified tolerance. In that case, we may have a very poor Newton update and thus the nonlinear iteration may take longer to converge. The other possibility for the differing numbers of

Table 1
Total nonlinear (NL) and linear iteration counts for solving the variably saturated flow test case with $40 \times 40 \times 55 = 88,000$ unknowns to 25 days with increasing variances

Preconditioner	Variance	NL iterations	Linear iterations	Compute time (s)
PFMG	0	41	148	946
SMG	0	39	137	1085
PFMG	5	68	232	1509
SMG	5	61	215	1672
PFMG	10	121	434	2848
SMG	10	83	291	2260
SMG	15	74	337	2384
SMG	20	88	406	2888
SMG	25	110	531	3729

nonlinear iterations is that if the linear system tolerance is on the borderline of what is required to get the most effective Newton update, a superior preconditioner may solve the linear system more precisely in the final iteration than an inferior preconditioner would allow. With both preconditioners, the linear system will be solved to the required tolerance, but one will lead to a better Newton update than the other. Once this happens for a single linear solve, the two solves will be on differing paths to the solution. Note that the solutions will agree to as many digits as are requested in the nonlinear system tolerance for both solves.

The next test was to ascertain solver robustness for varying correlation lengths. Table 2 shows results from running the test case to five days with a factor of eight increase in correlation lengths and a variance of 10. See Fig. 1 for plots of the permeability fields. As the permeability data becomes more correlated, we see that the problem becomes easier to solve. Thus, with a variance for which PFMG struggles for short correlation lengths, we can see PFMG succeed for the longer correlation lengths. In summary, we see that for highly correlated fields, the solvers perform similarly to how they performed for low variances. This is not a surprise. It should be noted, however, that when choosing a solver for a given problem, the variance and correlations should both be considered. Using one independent of the other to choose a solver may lead to choosing a solver that takes too much time and memory to solve the problem or, still worse, fails to converge at all.

The third test was to look at run times specifically for a case exhibiting physical anisotropy. For this case, we considered a problem with an inclined plane running through it. The problem domain was 30 m × 30 m × 30 m. Porosity was 30%. Initial conditions were taken as a constant pressure head set to −10 m. Boundary conditions were specified to be Dirichlet values in hydrostatic equilibrium with interpolated values on a line going from the front left corner of the domain bottom to the back right corner of the domain bottom. The pressure head value at the two corners were −100 m at the front left and 100 m at the back right. In all cases, relative permeability and saturation curves were evaluated with van Genuchten formulas [16]. Exact formulas were used with no curve-fitting. The parameters α and n were set to 0.1/m and 2, respectively. The residual saturation was 20% and the domain was considered saturated at 99%. We used a single time-step of 1day. Nonlinear iteration stopping criteria were the same as in the first test case.

We imposed a permeability of $k_{\mathrm{D}} = 1$ m per day at points not on the inclined plane and changed the permeability value, k_{P}, for points on the plane from 1 to 10^5. Fig. 2 shows a schematic of the permeability field, and Fig. 3 shows the run times for the two preconditioners. We see that as k_{P} gets larger, PFMG struggles

Table 2
Total nonlinear (NL) and linear iteration counts for solving the variably saturated flow test case with $40 \times 40 \times 55 = 88,000$ unknowns to five days with increasing correlation lengths and a variance of 10

Preconditioner	Correlation lengths	NL iterations	Linear iterations	Compute time (s)
PFMG	25, 50, 2.5	89	314	2246
SMG	25, 50, 2.5	81	189	1841
PFMG	50, 100, 5	105	334	2500
SMG	50, 100, 5	61	214	1760
PFMG	100, 200, 10	101	342	2683
SMG	100, 200, 10	62	169	1535
PFMG	200, 400, 20	38	136	936
SMG	200, 400, 20	38	137	1127

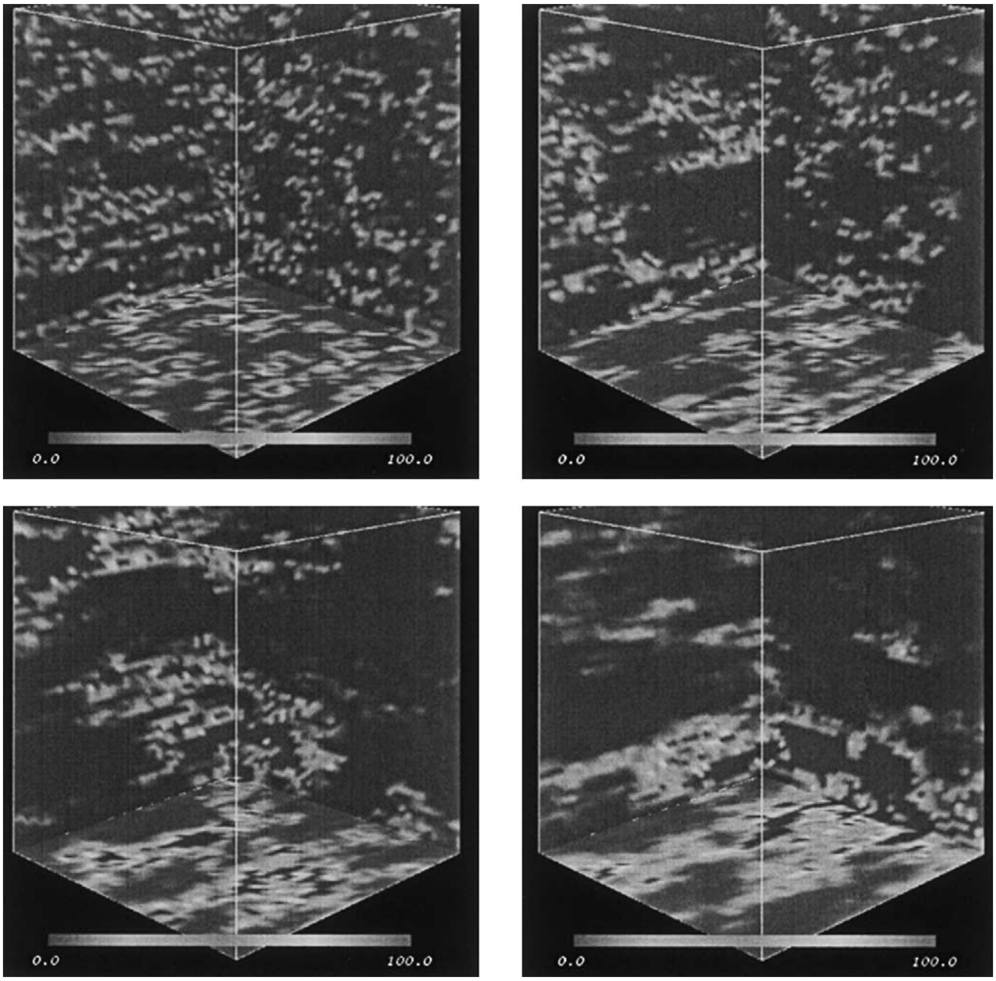

Fig. 1. Increasing correlation lengths ($\lambda_x, \lambda_y, \lambda_z$): (25 m, 50 m, 2.5 m) upper left, (50 m, 100 m, 5 m) upper right, (100 m, 200 m, 10 m) lower left, (200 m, 400 m, 20 m) lower right.

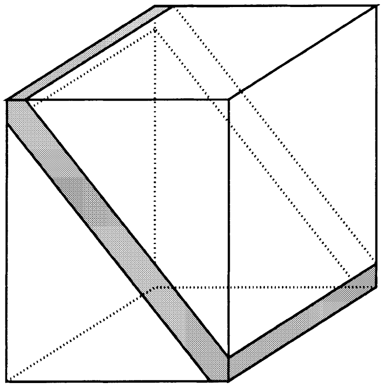

Fig. 2. Domain with inclined plane.

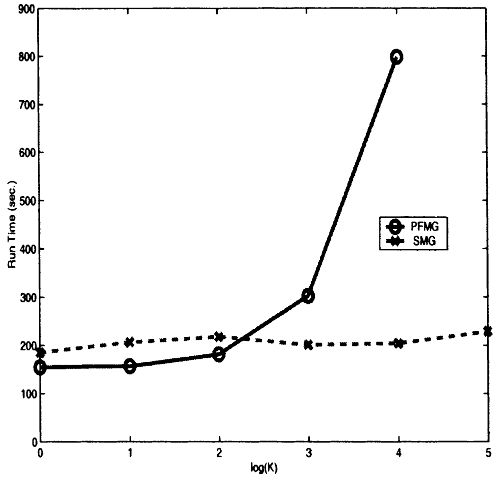

Fig. 3. Run times for varying permeability k_P on an inclined plane.

more and eventually fails to solve the problem for the 10^5 case. For low values, however, the PFMG preconditioner gives a faster solution.

Last, we investigated the parallel scalability of the nonlinear solution process. In the results we report the parallel scaled efficiencies defined as follows. Consider a global problem of size N distributed across P processors and let $T(N,P)$ be the run time for the code (or component of the code). The scaled efficiency is defined by

$E(n,p) = T(n,1)/T(pn,p)$. A scaled efficiency of one indicates perfect machine utilization, i.e., one can double the problem size without increasing the run time provided that one should also double the number of processors used in solving the problem.

Taking the LLNL-based test case, using a variance of 0, eight constant time steps of 0.0125 day, and all other problem data the same as in the first set of experiments, we performed a scaled speedup study with the two preconditioners. Results from this study are shown in Figs. 4 and 5. We added more unknowns by refining the domain in all three directions. Scaled efficiencies here are poor. Looking closely, we see that the poor performance is due to lack of scaling of the nonlinear function evaluation. As we decompose the domain in the z direction, we generate groupings of cell on the upper processors that contain more cells in the unsaturated zone than in processors with cells in the lower portion of the domain. As a result, the computation of the relative permeability and saturation in these upper domain cells is more expensive, and the nonlinear function is not well load-balanced. Decomposing the domain in only the x and y directions removes this imbalance, and we see that the scaled efficiencies are much improved in Figs. 6 and 7. This parallel study shows that modelers need to understand how best to utilize more resources in a given parallel machine, and be able to adjust the use of the machine to the physics of the problem being studied.

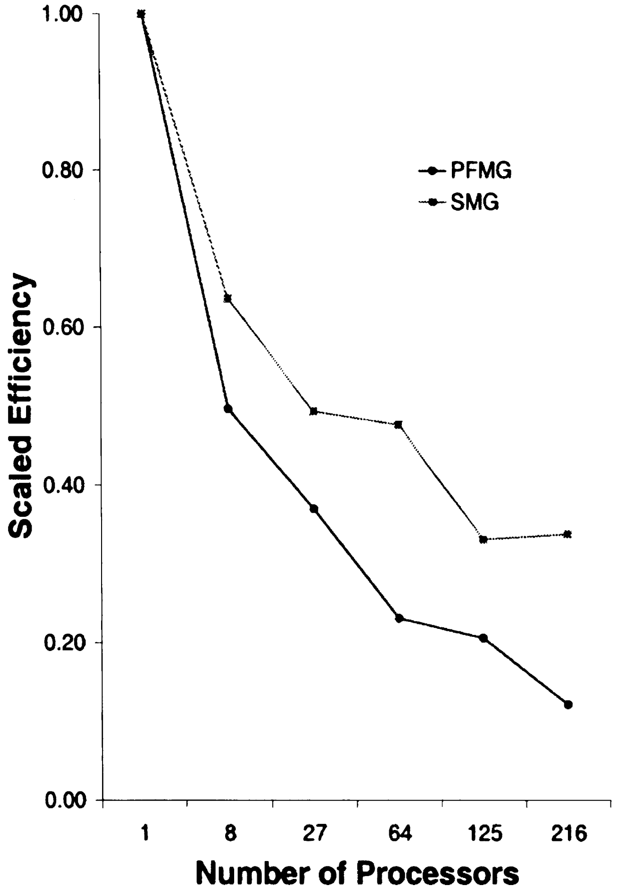

Fig. 4. Parallel scaled efficiencies for the full nonlinear solve using SMG and PFMG for decomposing in x, y, and z.

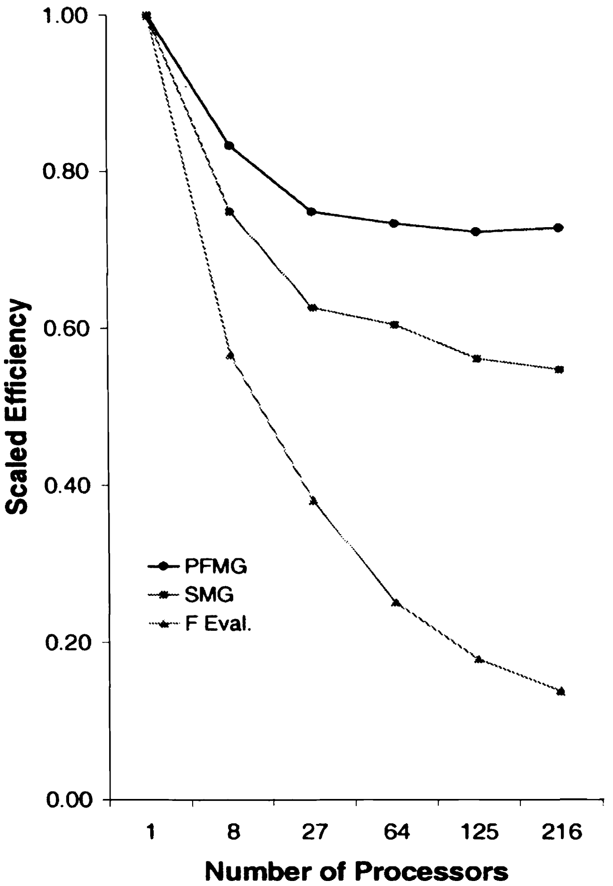

Fig. 5. Parallel scaled efficiencies for SMG, PFMG and the nonlinear function evaluation for decomposing in x, y, and z.

6. Conclusions

The results in this paper show that the Newton–Krylov-multigrid method can be an efficient and robust solver for large-scale, highly heterogeneous, variably saturated flow problems. The comparisons between the two multigrid solvers (SMG and PFMG) illustrate the balancing of efficiency and robustness. The PFMG solver is cheaper per linear iteration and, largely for this reason, faster than SMG when both methods work. However, for some difficult problems, SMG is more effective. The difficulty of the problem is influenced by the degree of heterogeneity, the correlation lengths, and the strength of anisotropies.

These results are important to consider when choosing the appropriate solver for a given physical problem. If the main axes of the permeability field do not align

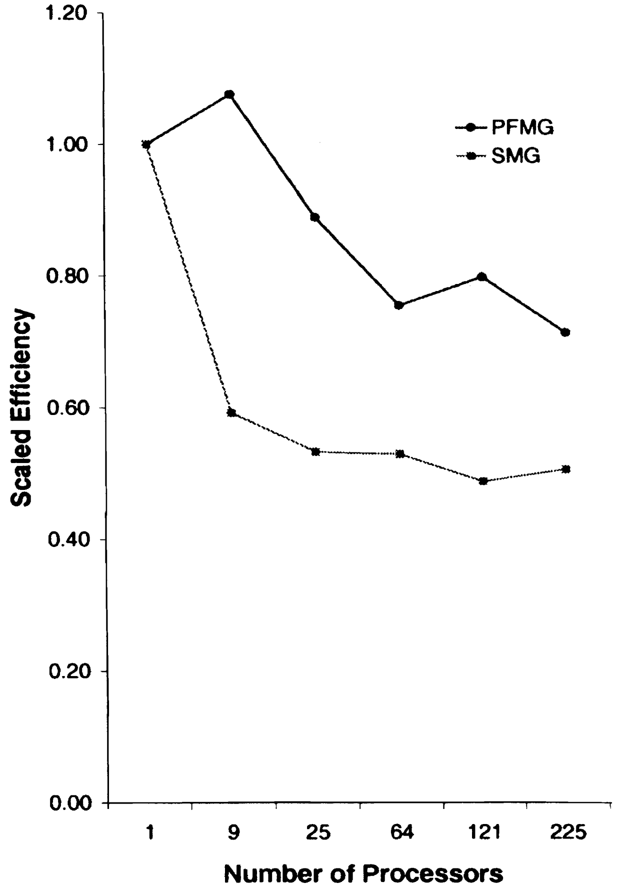

Fig. 6. Parallel scaled efficiencies for the full nonlinear solve using SMG and PFMG for decomposing in x and y.

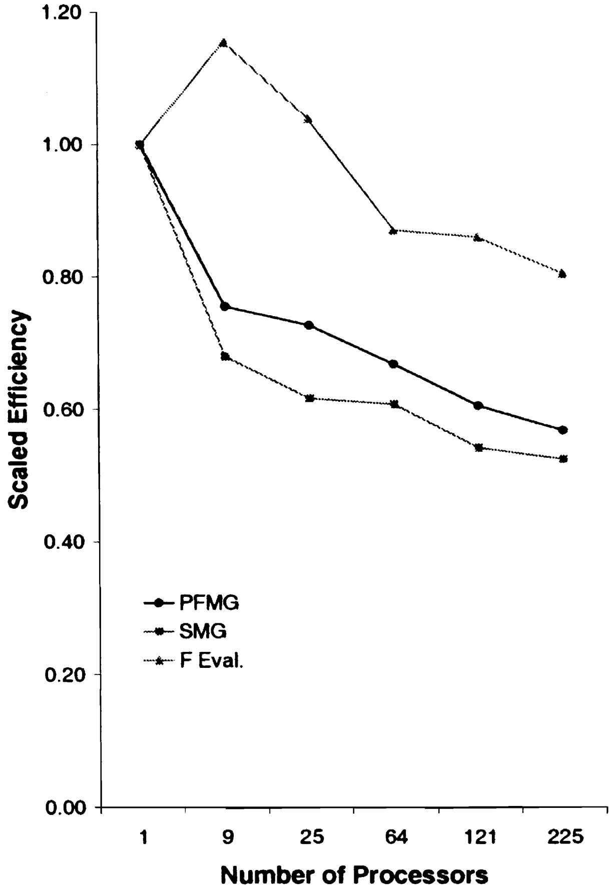

Fig. 7. Parallel scaled efficiencies for SMG, PFMG and the nonlinear function evaluation for decomposing in x and y.

with the numerical grid, then a complex solver such as SMG will be required. If the permeability is highly correlated, then a simpler solver such as PFMG should be effective even for higher variances.

Parallel load balancing of the problem physics must also be considered. A solution method can be computationally scalable for some parallel discretizations but seem not to be scalable for others if the wrong decomposition is used. In particular, the above results show that decomposing in the direction of water table fluctuations can lead to inefficiencies, and a better use of a parallel machine is to decompose the domain in directions orthogonal to these changes.

The results given here address solvers for the discrete, nonlinear systems of equations arising from the application of cell-centered discretization techniques to the mixed form of Richards' equation. Further work in this area would include investigation of Newton–Krylov-multigrid solvers for discrete systems arising from the discretization of transformed formulations of Richards' equation. Ref. [28] indicates these techniques can reduce nonlinear iteration counts over standard solvers. However, the performance of multigrid preconditioners for the resulting linear systems is unknown.

Results in this paper have shown solver responses to varying degrees of problem heterogeneity and anisotropy. We have not, however, addressed the case of spatially varying relative permeability and saturation curves. In some sense, this variation just adds another component to the heterogeneity of the problem. As a result, we expect the solvers' performances to compare similarly to what we have seen. Understanding the impact of this physical characteristic is another area of future work.

Acknowledgements

The authors thank Reed Maxwell of the Geosciences and Environmental Technologies Division at Lawrence Livermore National Laboratory for his enlightening discussions of subsurface modeling.

References

[1] Alcouffe RE, Brandt A, Dendy JE, Painter JW. The multi-grid methods for the diffusion equation with strongly discontinuous coefficients. SIAM J Sci Stat Comput 1981;2:430–54.

[2] Ashby SF, Falgout RD. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nucl Sci Eng 1996;124(1):145–59.

[3] Baldwin C, Brown PN, Falgout RD, Jones J, Graziani F. Iterative linear solvers in a 2d radiation-hydrodynamics code: methods and performance. J Comput Phys 1999;154:1–40.

[4] Briggs WF, Henson VE, McCormick SF. A multigrid tutorial. 2nd ed. Philadelphia, PA: SIAM; 2000.

[5] Brown PN. A local convergence theory for combined inexact-Netwon/finite-difference projection methods. SIAM J Numer Anal 1987;24:407–34.

[6] Brown PN, Falgout RD, Jones JE. Semicoarsening multigrid on distributed memory machines. SIAM J Sci Stat Comput 2000;21(5):1823–34.

[7] Brown PN, Saad Y. Hybrid krylov methods for nonlinear systems of equations. SIAM J Sci Stat Comput 1990;11:450–81.

[8] Celia MA, Bouloutas ET, Zarba RL. A general mass-conservative numerical solution for the unsaturated flow equation. Water Resour Res 1990;26(7):1483–96.

[9] Dendy JE. Black box multigrid. J Comput Phys 1982;48:366–86.

[10] Dendy JE, Ida MP, Rutledge JM. A semicoarsening multigrid algorithm for SIMD machines. SIAM J Sci Stat Comput 1992;13:1460–9.

[11] Dendy JE, McCormick SF, Ruge JW, Russell TF, Schaffer S. Multigrid methods for three-dimensional petroleum reservoir simulation. In: Proceedings of the 10th SPE Symposium on Reservoir Simulation. SPE, 1989.

[12] Dennis JE, Schnabel RB. Numerical methods for unconstrained optimization and nonlinear equations. Englewood Cliffs, NJ: Prentice-Hall; 1983.

[13] Eisenstat SC, Walker HF. Choosing the forcing terms in an inexact Newton method. SIAM J Sci Comput 1996;17(1):16–32.

[14] Forsyth PA, Kropinski MC. Monotonicity considerations for saturated–unsaturated flow. SIAM J Sci Comput 1997;18(5):1328–54.

[15] Forsyth PA, Wu YS, Pruess K. Robust numerical methods for saturated–unsaturated flow with dry initial conditions in heterogeneous media. Adv Water Resour 1995;18:25–38.

[16] van Genuchten MTh. A closed form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci Soc Am J 1980;44:892–8.

[17] Jones JE, Woodward CS. Preconditioning Newton–Krylov methods for variably saturated flow. In: Bentley LR, Sykes JF, Brebbia CA, Gray WG, Pinder GF, editors. Computational methods in water resources, vol. 1. Balkema: Rotterdam; 2000. p. 101–06.

[18] Kirkland MR, Hills RG, Wierenga PJ. Algorithms for solving Richards' equation for variably saturated soils. Water Resour Res 1992;28(8):2049–58.

[19] Miller CT, Williams GA, Kelley CT, Tocci MD. Robust solution of Richards' equation for nonuniform porous media. Water Resour Res 1998;34:2599–610.

[20] Richards LA. Capillary conduction of liquids through porous mediums. Physics 1931;1:318–33.

[21] Ross PJ, Bristow KL. Simulating water movement in layed and gradational soils using the Kirchhoff transform. Soil Sci Soc Am J 1990;54:1519–24.

[22] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986;7(3):856–69.

[23] Schaffer S. A semi-coarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. SIAM J Sci Comput 1998;20(1):228–42.

[24] Smith RA, Weiser A. Semicoarsening multigrid on a hypercube. SIAM J Sci Stat Comput 1992;13:1314–29.

[25] Tocci MD, Kelley CT, Miller CT. Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines. Adv Water Resour 1997;20(1):1–14.

[26] Tompson AFB, Ababou A, Gelhar LW. Implementation of the three-dimensional turning bands random field generator. Water Resour Res 1989;25:2227–43.

[27] Weiser A, Wheeler M. On convergence of block-centered finite differences for ellipitic problems. SIAM J Numer Anal 1988; 25:351–7.

[28] Williams GA, Miller CT, Kelley CT. Transformation approaches for simulating flow in variably saturated porous media. Water Resour Res 2000;36:923–34.

[29] Woodward CS, Dawson CN. Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media. SIAM J Numer Anal 2000;37(3):701–24.