Preprint
UCRL-JC-137011

Preconditioning Newton-Krylov Methods for Variably Saturated Flow

Jim E. Jones and Carol S. Woodward

This article appears in
Computational Methods in Water Resources, L. R. Bentley, J. F. Sykes, C. Brebbia, W. Gray, and G. F. Pinder, eds., vol. 1, Rotterdam, 2000, Balkema, pp. 101-106. (Proceedings of the XIII International Conference on Computational Methods in Water Resources, Calgary, Alberta, June 25-29, 2000)

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

January 2000

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes 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 the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
http://apollo.osti.gov/bridge/

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
http://www.ntis.gov/

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
http://www.llnl.gov/tid/Library.html

Preconditioning Newton-Krylov methods for variably saturated flow

J. E. Jones & C. S. Woodward
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, USA

ABSTRACT: In this paper, we compare the effectiveness of three preconditioning strategies in simulations of variably saturated flow. Using Richards' equation as our model, we solve the nonlinear system using a Newton-Krylov method. Since Krylov solvers can stagnate, resulting in slow convergence, we investigate different strategies of preconditioning the Jacobian system. Our work uses a multigrid method to solve the preconditioning systems, with three different approximations to the Jacobian matrix. One approximation lags the nonlinearities, the second results from discarding selected off-diagonal contributions, and the third matrix considered is the full Jacobian. Results indicate that although the Jacobian is more accurate, its usage as a preconditioning matrix should be limited, as it requires much more storage than the simpler approximations. Also, simply lagging the nonlinearities gives a preconditioning matrix that is almost as effective as the full Jacobian but much easier to compute.

1 INTRODUCTION

Accurate simulation of water resource management problems requires the solution of large problems with many spatial zones. In the case of variably saturated flow problems, we need to develop scalable and highly efficient algorithms for solving the large, nonlinear systems of equations that arise from the discretization of Richards' equation (Richards 1931).

One approach to solving these systems is to apply a Newton method requiring a linear Jacobian system solve at each iteration. Although Newton's method can have very fast convergence properties, it will run slowly if the linear system solver is not efficient. In this work, we solve the linear systems using a multigrid preconditioned Krylov method. Properly designed multigrid solvers are optimally efficient in that the work grows linearly with problem size while the convergence rate is constant. Most previous work on multigrid solvers, however, has been in the context of symmetric linear systems, and the Jacobian system for the discretized Richards' equation is nonsymmetric.

In this paper, we compare various strategies for solving the Jacobian system using multigrid. One approach is to base the multigrid preconditioner on a symmetric approximation to the Jacobian. We consider a symmetric approximation formed by deleting derivatives of relative permeability that lead to nonsymmetries, and also a symmetric approximation formed by lagging the nonlinearity in the Jacobian matrix. We compare the performance of methods based on these two symmetric approximations with that of a multigrid method based on the full nonsymmetric Jacobian. The two approximations have the obvious advantage of requiring less storage, but the true Jacobian has the advantage of capturing more of the problem's physics. We compare the efficiency of these various preconditioning strategies within the context of a Newton-Krylov method to solve the nonlinearities.

2 PROBLEM FORMULATION

We employ the mixed form of Richards' equation as our model of variably saturated flow,

$$\frac{\partial(s(p)\rho\phi)}{\partial t} - \nabla \cdot \left(\frac{k k_r(p)\rho}{\mu} (\nabla p - \rho g \nabla z) \right) = 0, \quad (1)$$

where $s(p)$ is water saturation, ρ is water density, ϕ is porosity of the medium, t is time, k is absolute permeability of the medium, $k_r(p)$ is relative permeability of water to air, μ is water viscosity, g is gravity and z is elevation.

Discretization is done for time with an implicit backward differencing scheme and for space with a cell-centered finite difference scheme. One-point upstream weighting is used for the face values of relative permeability and harmonic averaging for the absolute permeability. Applying these discretization schemes leads to a set of coupled discrete nonlinear equations that must be solved at each time step.

These discretization methods result in a coupled system of nonlinear equations that must be solved at each time step, where (considering just the x-direction for simplicity) the equation at each cell is given by,

$$F_i(p) = \Delta x_i \phi_i \rho(s(p_i^n) - s(p_i^{n-1})) \\ -\Delta t^n \left(\frac{U_{i+1/2}^n - U_{i-1/2}^n}{\Delta x_{i+1/2}} \right) = 0, \qquad (2)$$

where,

$$U_{i+1/2}^n = \left(\frac{k(x) k_r(p^n) \rho}{\mu} \right)_{i+1/2} \frac{p_{i+1}^n - p_i^n}{\Delta x_{i+1/2}}. \qquad (3)$$

3 SOLUTION METHOD

An inexact Newton-Krylov method is our nonlinear solver. Our use of this method is described by Woodward in (Woodward 1998) and includes the use of dynamic selection of linear system tolerances with a method of Eisenstat and Walker (Eisenstat & Walker 1996) and the line-search globalization strategy given by Dennis and Schnabel (Dennis & Schnabel 1983). We also use the restarted version of the GMRES Krylov iterative solver developed by Saad and Schultz (Saad & Schultz 1986) to solve the linear Jacobian systems. The main advantage of using a Krylov method is that it only requires knowledge of the Jacobian matrix, J, through matrix-vector products which can be approximated by,

$$J(p^k)v \approx \frac{F(p^k + \epsilon v) - F(p^k)}{\epsilon}, \qquad (4)$$

where p^k is the kth Newton iterate for the pressure, F is the nonlinear function, v is the vector against which we multiply J, and ϵ is a difference parameter usually dependent on the size of machine round-off.

The GMRES method can suffer from stagnation if the linear system is left unpreconditioned. In order to make more progress in reducing the residual at each iteration, we precondition the linear Jacobian system. 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.

3.1 *Multigrid preconditioner*

Previous work (Jones & Woodward 2000) has shown multigrid to be a robust and scalable method for solving the linear preconditioning systems (computing M^{-1}). Multigrid's chief advantage is that it is a scalable algorithm. Thus, when properly designed, the convergence rate of multigrid is independent of the size of the discretized system. 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).

Let $MU = B$ be the given linear system to solve, here the unknown U and right-hand side B are vectors defined on a logically rectangular grid. We will use an h superscript to denote quantities defined on the given grid, and a $2h$ superscript to denote quantities defined on a coarser grid. The multiple grids are used in a multigrid V-cycle, as follows,

$V(\nu_1, \nu_2)$-cycle

1. Perform pre-relaxation on $M^h U^h = B^h$ by executing ν_1 sweeps of a simple solver.
2. Set $B^{2h} = I_h^{2h}(B^h - M^h U^h)$.
3. Solve $M^{2h} U^{2h} = B^{2h}$ by recursion. If on the coarsest grid, solve this system exactly.
4. Correct $U^h \leftarrow U^h + I_{2h}^h U^{2h}$.
5. Perform post-relaxation on $M^h U^h = B^h$ by executing ν_2 sweeps of a simple solver.

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_h^{2h} used to transfer vectors from fine to coarse grids, and the coarse operator M^{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 modeling subsurface flow, the multigrid solver must be able to efficiently deal with anisotropies and widely variable permeability coefficients. In this work we use the semicoarsening multigrid (SMG) algorithm developed by Schaffer (Schaffer 1999). In the SMG algorithm, plane-wise red/black Gauss–Seidel relaxation is used. The values of the current approximation on the planes perpendicular to the z-axis (the xy-planes) are updated simultaneously to satisfy the equations on these planes. These plane solves are achieved by one 2D multigrid V-cycle. This relaxation provides robustness for anisotropic problems where the strong coupling is within the planes. Semicoarsening (i.e. coarsening the grid only in the direction perpendicular to the relaxation planes) 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_{2h}^{h}. It is built from the coefficients of the matrix M^h by following an approximate reduction method, see (Schaffer 1999) for details. The resulting interpolation operator is “operator dependent” in that the coefficients of M^h effect the interpolation weights. This provides robustness for problems with discontinuous coefficients. In our tests, we use the transpose of interpolation for restriction (i.e. $I_h^{2h} = (I_{2h}^{h})^T$) and form the coarse grid matrix by the Galerkin method (i.e. $M^{2h} = I_h^{2h} M^h I_{2h}^h$). These choices preserve symmetry, M^{2h} is symmetric if M^h is. However, for nonsymmetric M^h, other choices for restriction are possibly more appropriate, see (Bandy et al. 1998).

3.2 *Choice of preconditioning matrices*

The best choice of the preconditioning matrix, M, is unclear. In this work, we have compared three possibilities for this matrix. To best motivate these choices, we consider the differential form of the nonlinear system to be solved at time step n,

$$\begin{aligned} F(p^n) &= -\nabla\cdot(K(p^n)(\nabla p^n - \rho g \nabla z)) \\ &\quad + \frac{s(p^n)\phi\rho - s(p^{n-1})\phi\rho}{\Delta t^n} = 0, \end{aligned} \quad (5)$$

where $K(p^n) = \frac{k k_r(p^n)\rho}{\mu}$ is the nonlinear coefficient.

At Newton step $k+1$ for time step n, we need to solve the Jacobian system, $J(p^{n,k})h^k = -F(p^{n,k})$, for the pressure increment h^k, where $p^{n,k}$, is the kth Newton iterate for the pressure at time step n. The differential form of this system is derived by taking the Frechét derivative of the nonlinear function to get the action of the Jacobian on the vector h^k,

$$J(p^{n,k})h^k = \lim_{\epsilon\to 0} \frac{F(p^{n,k}+\epsilon h^k) - F(p^{n,k})}{\epsilon}, \quad (6)$$

which is just

$$\begin{aligned} \frac{1}{\Delta t^n} \quad & \lim_{\epsilon\to 0} \quad \frac{(s(p^{n,k}+\epsilon h^k) - s(p^{n,k}))\phi\rho}{\epsilon} \\ - \quad & \lim_{\epsilon\to 0} \quad \left(\frac{\nabla\cdot(K(p^{n,k}+\epsilon h^k)\nabla(p^{n,k}+\epsilon h^k))}{\epsilon} \right. \\ & \qquad \left. - \frac{\nabla\cdot(K(p^{n,k}+\epsilon h^k)\rho g\nabla z)}{\epsilon} \right) \\ + \quad & \lim_{\epsilon\to 0} \quad \frac{\nabla\cdot(K(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z))}{\epsilon}. \end{aligned} \quad (7)$$

Adding and substracting $\frac{\nabla\cdot(K(p^{n,k}+\epsilon h^k)\nabla p^{n,k})}{\epsilon}$ and then taking the limit in the derivative gives the following system,

$$\begin{aligned} & \frac{s'(p^{n,k})\phi\rho}{\Delta t^n} h^k - \nabla\cdot(K(p^{n,k})\nabla h^k) \\ & -\nabla\cdot(h^k K'(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z)) \\ & \qquad = -F(p^{n,k}), \end{aligned} \quad (8)$$

where s' and K' are the derivatives of the saturation and the nonlinear coefficient, respectively.

For the discretization methods used in this work and discussed in Section 2, as long as interface averaging is done consistently, an equivalence can be shown between the discretized form of this Jacobian system and the Jacobian matrix arising from differentiating the entries of the nonlinear function, F. Thus, we can discuss the Jacobian and its preconditioner in this differential form where the operator is easier to understand.

Note that the second left-hand-side term of (8) is an elliptic term operating on the unknown h^k. The third left-hand-side term is first order in h^k, involves derivatives of the nonlinear coefficient function, and leads to nonsymmetries in the discrete matrix. Our choices of preconditioning matrices differ in the way that they approximate the third left-hand-side term of (8).

Our first preconditioning matrix choice is to approximate the full Jacobian system (8) by taking all terms except the third left-hand-side one. This is similar to applying a single iteration of a fixed point scheme to the nonlinear system. We call this the Picard approach.

Our second choice of preconditioning matrix is motivated by expanding the third left-hand-side term in (8) as,

$$\begin{aligned} & -\nabla\cdot(h^k K'(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z)) = \\ & \quad -K'(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z)\cdot\nabla h^k \\ & \quad -\nabla\cdot(K'(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z))h^k. \end{aligned} \quad (9)$$

When discretization is applied, the first right-hand-side term of this expansion leads to off-diagonal elements of the linear system matrix, while the second term only leads to diagonal elements. Our second choice in preconditioning matrix arises from neglecting the first right-hand-side term in this expansion and approximating the Jacobian system with,

$$\begin{aligned} J(p^{n,k})h^k \approx & \frac{s'(p^{n,k})\phi\rho}{\Delta t^n} h^k - \nabla\cdot(K(p^{n,k})\nabla h^k) \\ & -\nabla\cdot(K'(p^{n,k})(\nabla p^{n,k} - \rho g\nabla z))h^k. \end{aligned} \quad (10)$$

We call this the symmetric approximation approach. Note that because $K' \geq 0$ for all p, and since we are using upstream weighting for the relative permeabilities, we know that the third right-hand-side term of (10) leads to a positive contribution to the matrix diagonal.

These first two preconditioning choices lead to symmetric preconditioning matrices for M. Thus, they require significantly less storage than the full Jacobian. The third choice of preconditioning matrix that we consider is just to use the full Jacobian, and not an approximation to it. However, this choice requires full matrix storage and more time to compute. In addition, much of the multigrid literature has been geared toward the solution of symmetric matrices, and solver heuristics may not perform as well for the full Jacobian.

4 NUMERICAL RESULTS

In order to test the relative effectiveness of the three preconditioning choices outlined above, we have included these preconditioning matrices in the variably saturated flow option of the ParFlow three-dimensional, parallel porous media flow simulator.

We consider a test case with a saturated zone at the bottom of the domain and with an inflow flux from the right. We first ran some comparisons with an essentially two-dimensional case (no heterogeneity in the permeability field) and then with a three-dimensional case where we applied some heterogeneity using the turning bands algorithm (Tompson et al. 1989). The specifics of the test case are as follows. The domain is a 100 cm cube, and the mean permeability is 4.0cm^2. Heterogeneous cases have a standard deviation of 3.0cm^2, correlation lengths in the x and y directions of 50cm and in the z direction of 20cm. Constant head boundary conditions of 20cm and –20cm, respectively, are specified on the left and right faces with no flow conditions everywhere else. An initial condition of –20cm is taken over the entire domain. The residual saturation is $S_{res} = 0.2$ and the domain is considered fully saturated when saturation is 0.99. Porosity is taken to be 30%. To run the solvers over a variation of time step sizes, we started with a time step of 0.0005day and doubled the step size after each step, eventually completing eleven steps to get to the final time of 1day. The homogeneous cases used $30 \times 10 \times 30$ grids; the heterogeneous cases used $30 \times 30 \times 30$ grids.

We have used the vanGenuchten (vanGenuchten 1980) curves for the specification of relative permeabilities and saturations and include two cases. For both cases, we specify n as 3.0. In one case, we specify the α parameter as 0.02 and in the other $\alpha = 0.2$. Figures 1 and 2 show the curves for the two cases. We see that for the $\alpha = 0.02$ case, the k_r and s curves look like broadened s-curves. For the $\alpha = 0.2$ case, the curves are very steep. In general, we would expect the first case to be easier to solve as it gives rise to a diffusive solution and a more slowly moving time-dependent behavior. We expect the second case to be harder as it gives rise to a much more quickly changing solution. In the second case, the nonlinearities of the problem should be more difficult for the solvers.

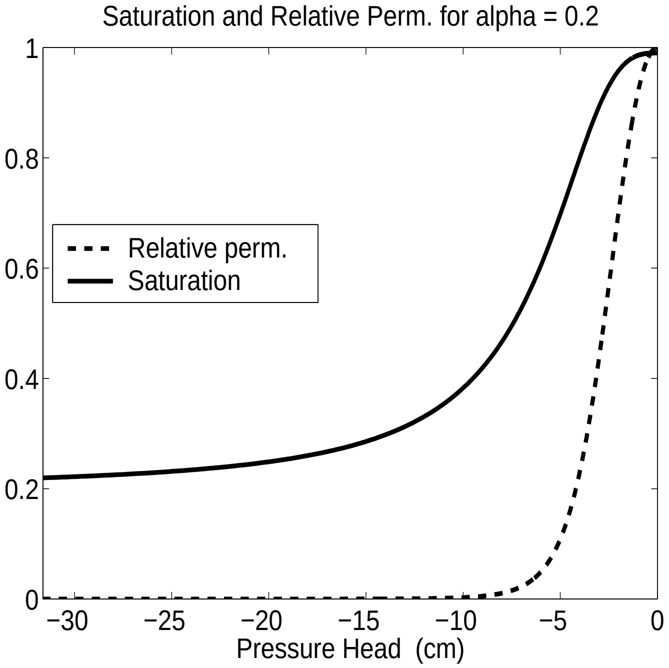

Figure 1: Relative permeability and saturation curves for $\alpha = 0.02$, $n = 3.0$.

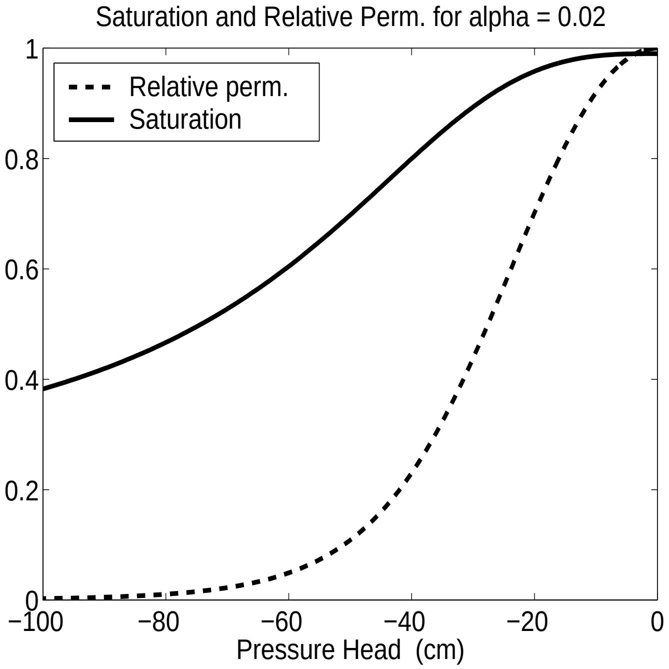

Figure 2: Relative permeability and saturation curves for $\alpha = 0.2$, $n = 3.0$.

We allow one multigrid V-cycle per preconditioner call, a maximum of 30 nonlinear iterations and a maximum of 30 Krylov vectors. All runs were done on a SUN Sparc Ultra 10 workstation at Lawrence Livermore National Laboratory.

The results for the homogeneous cases are summarized in Tables 1 and 2. We report the number of linear and nonlinear iterations taken at each time step for each of the three preconditioning strategies: full Jacobian, symmetric approximation, and Picard approach. The tables also report the total computing time.

Table 1: Homogeneous case with $\alpha = 0.2$. LI = linear iterations, and NI = nonlinear iterations.

Time	Jaccobian		Symmetric		Picard	
(day)	LI	NI	LI	NI	LI	NI
0.0005	7	4	7	4	7	4
0.0015	7	5	7	5	7	5
0.0035	9	5	9	5	9	5
0.0075	8	6	8	6	9	6
0.0115	9	6	9	6	9	6
0.0315	8	5	9	5	9	5
0.0635	10	5	13	5	13	5
0.1275	10	7	15	7	15	7
0.2555	12	8	20	6	27	7
0.5115	17	9	40	9	23	9
1.0000	25	13	48	12	53	10
Totals	122	73	185	70	201	69
Run Time	242 sec		264 sec		270 sec	

Table 2: Homogeneous case with $\alpha = 0.02$. LI = linear iterations, and NI = nonlinear iterations.

Time	Jacobian		Symmetric		Picard	
(day)	LI	NI	LI	NI	LI	NI
0.0005	7	4	7	4	7	4
0.0015	8	4	8	4	8	4
0.0035	8	4	8	4	8	4
0.0075	9	4	8	4	7	4
0.0115	9	4	8	4	7	4
0.0315	9	4	11	4	11	4
0.0635	8	4	15	5	14	5
0.1275	10	4	17	5	16	5
0.2555	13	5	22	5	19	5
0.5115	13	5	29	6	24	6
1.0000	17	6	34	6	27	7
Totals	111	48	167	51	148	52
Run Time	172 sec		203 sec		191 sec	

For the case $\alpha = 0.2$, the numbers of nonlinear and linear iterations are plotted in Figures 3 and 4. Here we see that as the time step size grows, the number of nonlinear and linear iterations grows. The growth in the number of linear iterations is particularly dramatic with the symmetric approximation and the Picard approach; as the time steps grow, the linear systems become less diagonally dominant and thus harder to solve. As one would expect, the full Jacobian is the best preconditioner (both in terms of number of linear iterations and computing time) followed by the symmetric approximation and the Picard approach, in that order. Changing α to 0.02, we see that the nonlinear problem becomes easier. The numbers of nonlinear and linear iterations, and total computing time are all reduced. This agrees with our intuition mentioned above in discussing the shapes of the Van Genuchten curves. Note that in this case, the Picard approach outperforms the symmetric approximation.

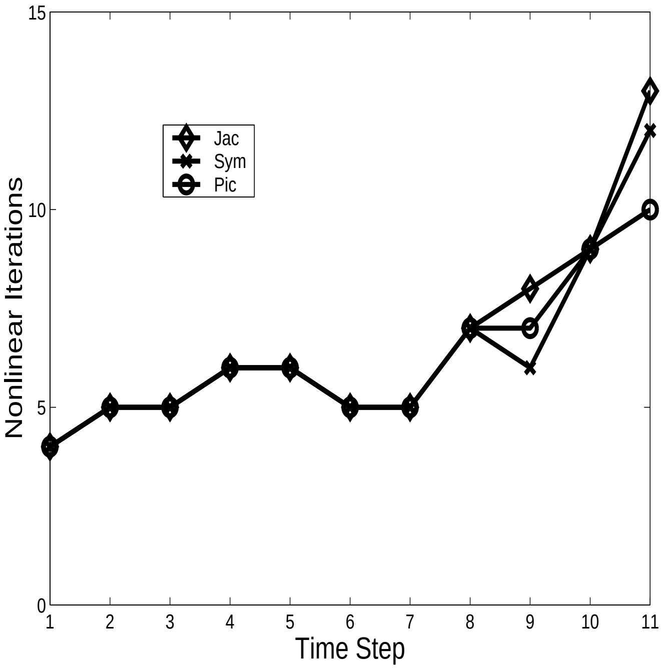

Figure 3: Number of nonlinear iterations at each time step

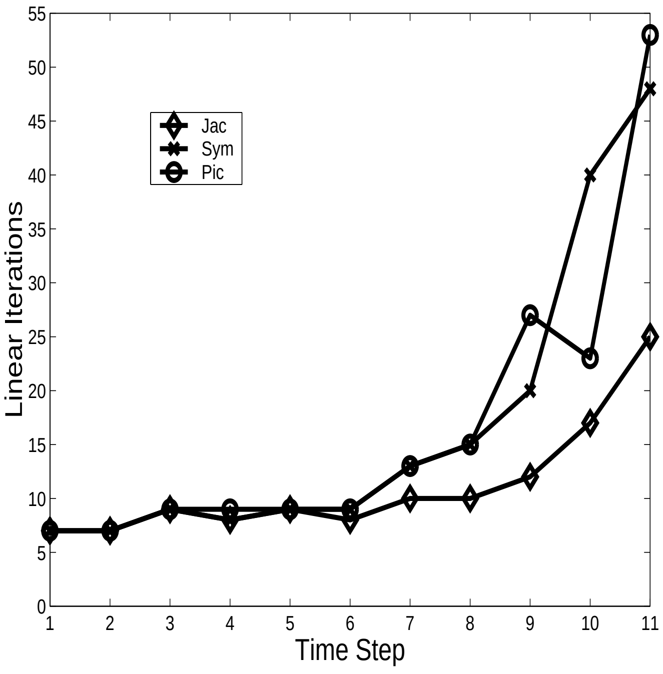

Figure 4: Number of linear iterations at each time step

The results for the heterogeneous cases are summarized in Tables 3 and 4. Here, we report only the total number of linear and nonlinear iterations taken and the total computing time. In terms of overall run time, the results using the full Jacobian and the Picard approach are quite close.

Table 3: Heterogeneous case with $\alpha = 0.2$. LI = linear iterations, and NI = nonlinear iterations.

	Jacobian		Symmetric		Picard	
	LI	NI	LI	NI	LI	NI
Totals	182	102	359	155	315	91
Run Time	954 sec		1495 sec		943 sec	

Table 4: Heterogeneous case with $\alpha = 0.02$. LI = linear iterations, and NI = nonlinear iterations.

	Jacobian		Symmetric		Picard	
	LI	NI	LI	NI	LI	NI
Totals	162	63	232	66	197	64
Run Time	611 sec		688 sec		622 sec	

For this test case, the results using the symmetric approximation are considerably worse than the other two, especially in the case $\alpha = 0.2$. For this value, the code reached the maximum number of nonlinear iterations (30) for the tenth time step and then reduced the size of the time step and completed two more steps before getting to the final time. We are unsure as to why this matrix approximation is inferior in this test problem. However, we believe that the poor performance is related to the fact that we are including a term that inflates the matrix diagonal without making a corresponding increase in the off-diagonals. Determination of why this approximation performs the way it does and perhaps finding a better symmetric approximation to the Jacobian are the subjects of current and future research.

5 CONCLUSIONS

The numerical tests described above, although limited in scope, show that the Jacobian matrix appears to be an effective preconditioning matrix. However, compute times for this choice are not much better than for using the Picard approach, which requires just over half the storage of the full Jacobian. In addition, use of the Picard approach does not require computing derivatives of relative permeabilities at any time in the solution process. In cases where these values are obtained by table lookups, this avoidance of derivative computation can reduce potential inaccuracies as well as saving compute time.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

REFERENCES

Bandy, V., Dendy, J. E., & Spangenberg, W. H. (1998). Some multigrid algorithms for elliptic problems on data parallel machines. *SIAM J. Sci. Stat. Comput. 19*, 74–86.

Dennis, J. E. & Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Eisenstat, S. C. & Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comp. 17*(1), 16–32.

Jones, J. E. & Woodward, C. S. (2000). Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Manuscript in preparation.

Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics 1*, 318–333.

Saad, Y. & Schultz, M. H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput. 7*(3), 856–869.

Schaffer, S. (1999). A semi-coarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. *SIAM J. Sci. Comp. 20*, 228–242.

Tompson, A., Ababou, A., & Gelhar, L. W. (1989). Implementation of the three-dimensional turning bands random field generator. *Water Resour. Res. 25*, 2227–2243.

vanGenuchten, M. T. (1980). A closed form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J. 44*, 892–898.

Woodward, C. S. (1998). A Newton-Krylov-multigrid solver for variably saturated flow problems. In *Proc. of the Twelfth International Conference on Computational Methods in Water Resources*, Volume 2, Southampton, pp. 609–616. Computational Mechanics Publications.

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551