Efficient Steady-State Solution Techniques for Variably Saturated Groundwater Flow

Matthew W. Farthing, [a,*] Christopher E. Kees, [b]

Todd S. Coffey, [c] C.T. Kelley [b] Cass T. Miller [a]

[a] *Center for the Advanced Study of the Environment, Department of Environmental Sciences and Engineering, University of North Carolina, Chapel Hill, North Carolina 27599-7431, USA*

[b] *Center for Research in Scientific Computation, Department of Mathematics, North Carolina State University, Raleigh, North Carolina, 27695-8205, USA*

[c] *Mathematical Information and Computational Sciences, Sandia National Laboratories, Albuquerque, New Mexico, 87185-1110, USA*

Abstract

We consider the simulation of steady-state variably saturated groundwater flow using Richards' equation (RE). The difficulties associated with solving RE numerically are well known. Most discretization approaches for RE lead to nonlinear systems that are large and difficult to solve. The solution of nonlinear systems for steady-state problems can be particularly challenging, since a good initial guess for the steady-state solution is often hard to obtain, and the resulting linear systems may be poorly scaled. Common approaches like Picard iteration or variations of Newton's method have their advantages but perform poorly with standard globalization techniques under certain conditions.

Pseudo-transient continuation has been used in computational fluid dynamics for

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
30 OCT 2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Efficient Steady-State Solution Techniques for Variably Saturated Groundwater Flow	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **North Carolina State University,Center for Research in Scientific Computation,Raleigh,NC,27695-8205**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**47**	

some time to obtain steady-state solutions for problems in which Newton's method with standard line-search strategies fails. It combines aspects of backward Euler time integration and Newton's method to select intermediate estimates of the steady-state solution. Here, we examine the use of pseudo-transient continuation as well as Newton's method combined with standard globalization techniques for steady-state problems in heterogeneous domains. We investigate the methods' performance with direct and preconditioned Krylov iterative linear solvers. We then make recommendations for robust and efficient approaches to obtain steady-state solutions for RE under a range of conditions.

Notation

Roman Letters

A	accumulation term for pressure head form of RE
$\mathbf{A}$	accumulation term contribution to Jacobian
$\mathbf{J}$	Jacobian
C	scaling factor for Dirichlet boundary conditions
$\mathbf{F}$	nonlinear function for DAE formulation, semi-discrete
$\mathbf{G}$	nonlinear function for DAE formulation, discrete
K_s	saturated hydraulic conductivity
K_s^s	surface saturated hydraulic conductivity

* Corresponding author
Email addresses: matthew_farthing@unc.edu (Matthew W. Farthing,), chris_kees@ncsu.edu (Christopher E. Kees,), tscoffe@sandia.gov (Todd S. Coffey,), tim_kelley@ncsu.edu (C.T. Kelley), casey_miller@unc.edu (Cass T. Miller).

K_e	effective hydraulic conductivity
N_{max}	maximum number of iterations for nonlinear solution methods
$\mathbf{O}_d$	discrete spatial operator
$\mathbf{R}$	discrete spatial operator and source term
S_e	effective saturation
T	extent of temporal domain
X_d	extent of spatial domain along x_d axis
$\mathbf{f}$	source term for the aqueous phase evaluated at cell centers
f	source term for the aqueous phase
$\mathbf{g}$	gravitational vector
$\mathbf{g}_u$	unit vector, $\mathbf{g}/\|\mathbf{g}\|$
k_i	intrinsic permeability
k_r	relative permeability
m_v	parameter for VGM
$\mathbf{n}$	unit outward normal for Ω
n_e	total number of nodes
n_v	parameter for VGM
n_{xd}	number of nodes along x_d axis
p	pressure of the aqueous phase
t	time coordinate
$\mathbf{u}$	mass flux
u^b	Neumann boundary value
u^r	precipitation rate
$\mathbf{y}$	variable for DAE formulation
$\mathbf{y}'$	temporal derivative for DAE formulation
y_{min}	lower bound for solution in test for evaluation error
y_{max}	upper bound for solution in test for evaluation error

Greek Letters

Δt	time step for Ψtc methods
Δt_{max}	maximum time step for Ψtc methods
Δx_d	spatial increment in x_d direction
$\Delta \mathbf{y}$	Newton increment
Γ	boundary of physical domain
Γ_D	portion of Γ for which Dirichlet boundary conditions are set
Γ_N	portion of Γ for which Neumann boundary conditions are set
Ω	physical domain
α_v	parameter for VGM
β	compressibility of the aqueous phase
ϵ_c	switching tolerance in hybrid Newton-Picard method
ϵ_s	sufficient decrease parameter for Armijo line search
θ	volume fraction of the aqueous phase
θ_r	residual volumetric water content
θ_s	saturated volumetric water content
λ	line-search scaling factor
λ_+	intermediate scaling factor for quadratic line search
μ	viscosity of the aqueous phase
ϱ	density of the aqueous phase
ρ	normalized density of the aqueous phase
σ_{min}	bound parameter for quadratic line search
σ_{max}	bound parameter for quadratic line search
τ	TTE parameter for truncation error estimate bound
ψ	pressure head
ψ^b	Dirichlet boundary value

ψ^0	initial condition for ψ

Subscripts and Superscripts

d	coordinate axis identifier (subscript)
i	cell qualifier along x_1 axis (subscript)
j	cell qualifier along x_2 axis (subscript)
l	global identifier for solution unknowns (subscript)
n	time level identifier (superscript)
s_n	nonlinear iteration index (superscript)

Abbreviations

Clock	wall clock time (seconds)
DAE	differential algebraic equation
Feval	function evaluation
HAS	two-level hybrid additive Schwarz domain decomposition preconditioner
Jeval	Jacobian evaluation
LF	linear iteration failure
LI	linear iteration
LU-B	banded LU decomposition from LAPACK
LU-S	sparse LU decomposition from SPOOLES (version 2.2)
NILS	Newton's method with a quadratic line search
NLF	nonlinear iteration failure
NLI	nonlinear iteration
ODE	ordinary differential equation

PIH	Newton-Picard hybrid approach
RE	Richards' equation
SER	switched evolution relaxation Ψtc approach
Steps	attempted iterations for method
TTE	temporal truncation errror Ψtc approach
VGM	combined van Genuchten and Mualem p-S-k relation
p-S-k	pressure-saturation-relative permeability constitutive relation
Ψtc	pseudo-transient continuation

1 Introduction

Variably saturated groundwater flow is commonly modeled using Richards' equation (RE) along with a set of constitutive relations describing the interdependence among fluid pressures, saturations, and relative permeabilities (p-S-k relations). While analytical and semi-analytical approximations for variably saturated flow exist, these are valid for limited sets of auxiliary conditions and domains [30]. As a result, significant effort has focused on developing robust techniques for solving RE numerically [9, 38, 29, 32, 27, 34, 36, 37]. Obtaining solutions for RE for many realistic physical conditions remains a challenge. Infiltration problems are often characterized by sharp fronts in both space and time. Steady-state solution is often nontrivial as well, since the volume fraction can vary steeply for problems with realistic boundary conditions and heterogeneous porous media.

It is the nature of the nonlinearities introduced through standard p-S-k relations like the van Genuchten [35] and Mualem [28] models that accounts for the majority of the difficulties associated with solving RE. A number of issues associ-

ated with resolving this nonlinear behavior have received attention. These include solution variable transformations [4, 36], the evaluation of p-S-k relations [33], approximation of interface conductivities in standard low-order spatial discretizations [38, 27], choice of dependent variable [37], and time discretization approach [9, 31, 33, 19]. Since the majority of time discretizations are implicit, both transient and steady-state problems typically lead to the solution of a system of discrete nonlinear equations. These systems can be quite large and difficult to solve, especially for problems with heterogeneous domains in two and three spatial dimensions.

The nonlinear system solution method used can then have a strong impact on the overall success of a simulator for RE. The most common approaches have been Picard iteration [9, 10, 19] or a variant of Newton's method [34, 37, 21]. Each of these methods has its strengths and weaknesses, and several works have compared the robustness and efficiency of Picard and Newton approaches for a variety of problems [29, 25, 27, 8]. Newton's method combined with globalization techniques like a line search, reduction of time step (backtracking), or the use of Picard iteration to obtain an initial guess has proven more reliable and efficient than Picard iteration for several problems [29, 27]. However, both Newton and Picard approaches have been shown to perform poorly under certain conditions. The solution of nonlinear systems can be particularly difficult for steady-state problems due to the increased difficulty of obtaining a good initial guess for the steady-state solution. In addition, the scaling of the linearized systems is typically worse [15], and there is no longer recourse to reducing time steps when convergence breaks down [29].

Pseudo-transient continuation (Ψtc) has been used in computational fluid dynamics for some time to obtain steady-state solutions for problems in which

Newton's method with standard line-search strategies fails [23, 24, 11]. Ψtc combines features of backward Euler time integration and Newton's method with the idea of using information about the transient physical problem to guide selection of intermediate iterates to the steady-state solution [15]. Since it uses information from a time-dependent problem, Ψtc can be more robust than standard line-search strategies, while incurring less computational expense than full integration of the transient problem [23].

In this work, we seek effective techniques for obtaining steady-state solutions to RE. We examine the use of Ψtc methods as well as Newton's method combined with standard globalization techniques for steady-state problems in homogeneous and heterogeneous domains. We investigate the methods' performance with both direct and preconditioned Krylov iterative linear solvers. We then make recommendations for robust and efficient approaches to obtain steady-state solutions for RE under various conditions.

2 Background

Historically, Picard iteration has been the most common nonlinear solution method used for RE [9, 31]. It's appeal can be traced to the fact that it is simple to implement, since it does not involve derivatives of terms involving the p-S-k relations and produces symmetric linear systems [29]. While Picard iteration is still used [10, 19], Newton and inexact Newton approaches have become significantly more common [14, 18, 21], particularly for more realistic multidimensional problems. An inexact Newton approach can be thought of as Newton's method where the equation for the Newton update is satisfied only approximately [34]. This may be due to the fact that the Newton update

is solved only approximately using, for example, a Krylov-type iterative linear solver [18]. Or, only an approximate Jacobian may be used in the Newton update. Common approaches for approximating the full Jacobian are to use finite differences (a numerical Jacobian) [14] or to lag its evaluation over a number of iterates (chord or modified Newton's method) [33].

While these approaches may be conceptually straightforward, a number of subtleties often arise [22]. For instance, the choice of stopping criteria and the associated tolerances for both the nonlinear and linear solvers, the frequency of Jacobian evaluations, and the increment used in evaluating numerical Jacobians can all play a significant role in the success of an approach for a given problem [29, 25, 27, 8, 34]. Since the convergence of Newton's method can be quite sensitive to the quality of initial guess, supplementing a Newton approach with strategies like a line search and backtracking has improved nonlinear solver performance for some problems [27]. Line-search techniques typically select a fraction of the full Newton correction to update the current iterate with the step chosen to insure that the updated solution represents a sufficient decrease in the nonlinear residual [18, 11]. For transient problems, methods commonly adjust the time step based on nonlinear solver performance. Specifically, in the case of poor nonlinear solver performance, the time step is often reduced [31, 33, 8]. This can improve the scaling of the Jacobian, since the reciprocal of the time step typically appears on the diagonal [15], and can improve the quality of the initial guess obtained from the solution at the previous time step [29]. Another strategy designed to address Newton approaches' sensitivity to initial guess is a hybrid Newton-Picard algorithm which uses a Picard iteration initially until a certain level of convergence is reached and then switches to Newton's method [29, 8].

Like the hybrid Newton-Picard algorithm, Ψtc attempts to combine the rapid convergence of Newton's method near the steady-state solution with a more robust iteration far from the solution. Ψtc employs a time-stepping approach when the iteration is far from the steady state, and thus exploits the relationship between the nonlinear system for the steady-state problem and the initial value problem from which it is derived. The time-stepping algorithm used in Ψtc has properties similar to forward Euler with simple heuristics, which makes it both unstable and inaccurate as a time-integration method for stiff ordinary differential equations (ODE's). Nevertheless, Ψtc can effectively maintain certain important qualities of the transient solution of some problems such as enforcing a CFL condition, and has been shown to be effective in reaching the steady state for models with discontinuous transient phenomena such as shocks. While RE does not exhibit shocks under physically realistic choices of parameters, it does exhibit sharp fronts due to the nonlinearities as well as steep gradients due to discontinuities in the porous media properties.

3 Approach

3.1 Formulation

RE can be formulated in a number of ways. We begin with an expression for the conservation of mass for the aqueous phase in an air-water system where the solid phase is assumed immobile, and interphase mass transfer is neglected. Combining this with the standard extension of Darcy's law to variable satu-

rated conditions [26] gives

$$\frac{\partial(\rho\theta)}{\partial t} = \nabla \cdot \rho K_e(\nabla\psi - \rho\mathbf{g}_u) + f \qquad \text{in } \Omega \times [0,T] \tag{1}$$

with

$$\begin{aligned}
\varrho &= \varrho_0 e^{\beta(p-p_0)} \\
\rho &= \varrho/\varrho_0 \\
\psi &= \tfrac{p}{\varrho_0\|\mathbf{g}\|} \\
\mathbf{g}_u &= \tfrac{\mathbf{g}}{\|\mathbf{g}\|} \\
K_s &= \tfrac{\varrho_0\|\mathbf{g}\|k_i}{\mu} \\
K_e &= k_r(\psi)K_s
\end{aligned} \tag{2}$$

Here p, ϱ, and μ are the pressure, density, and viscosity of the aqueous phase, θ is the volume fraction, and f is a source term for the aqueous phase. ϱ_0 is the density at p_0, β is the compressibility of the aqueous phase, ψ is the pressure head, and $\mathbf{g}$ is a vector accounting for the acceleration of gravity. K_e is the effective hydraulic conductivity, K_s is the saturated hydraulic conductivity, and k_i is the intrinsic permeability of the porous medium. $\Omega \subset \mathbb{R}^2$ is the spatial domain, and $[0,T]$ is the temporal domain.

The auxiliary conditions for eqn (1) are given by

$$\psi = \psi^b \qquad \text{on } \Gamma_D, t \in [0,T] \tag{3}$$
$$\mathbf{u}\cdot\mathbf{n} = u^b \qquad \text{on } \Gamma_N, t \in [0,T] \tag{4}$$
$$\psi = \psi^0 \qquad \text{in } \Omega, t = 0 \tag{5}$$

where

$$\mathbf{u} = -\rho K_e(\nabla\psi - \rho\mathbf{g}_u) \tag{6}$$

is the mass flux, ψ^b, and u^b are boundary condition functions. ψ^0 is the initial condition, and $\mathbf{n}$ is the unit outward normal for Ω. We have also set $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$.

The steady-state form of eqn (1) is

$$-\nabla \cdot \rho K_e(\nabla\psi - \rho\mathbf{g}_u) = f \qquad \text{in } \Omega \tag{7}$$

with auxiliary data given by eqns (3) and (4) without temporal dependence.

We use non-hysteretic forms of the p-S-k relations of van Genuchten [35] and Mualem [28] (VGM). For $\psi < 0$, these are given by

$$S_e = \frac{(\theta - \theta_r)}{(\theta_s - \theta_r)} \tag{8}$$

$$S_e = [1 + (\alpha_v|\psi|)^{n_v}]^{-m_v} \tag{9}$$

$$k_r = \sqrt{S_e}\left[1 - (1 - S_e^{1/m_v})^{m_v}\right]^2 \tag{10}$$

where S_e is the effective saturation, θ_r is the residual volumetric water content, θ_s is the saturated volumetric water content, α_v is a parameter related to the mean pore size, n_v is a parameter related to the uniformity of the soil pore-size distribution, and $m_v = 1 - 1/n_v$. For $\psi \geq 0$, the porous medium is fully saturated, and eqns (8)–(10) revert to

$$S_e = 1 \tag{11}$$

$$k_r = 1 \tag{12}$$

The pressure head form of RE is obtained from eqn (1) by applying the chain rule to the left hand side

$$A(\psi)\frac{\partial\psi}{\partial t} = \nabla \cdot \rho K_e(\nabla\psi - \rho\mathbf{g}_u) + f$$

$$A(\psi) = \theta \frac{\partial \rho}{\partial \psi} + \rho \frac{\partial \theta}{\partial \psi} \tag{13}$$

3.2 Spatial discretization

We consider a discretization of $\Omega = [0, X_1] \times [0, X_2] \subset \mathbb{R}^2$ into a regular, orthogonal grid with $n_e = n_{x1} \cdot n_{x2}$ nodes with $\Delta x_d = X_d/(n_{xd} - 1),$ for $d = 1, 2$. We apply a cell-centered finite difference approximation to the right hand side of eqn (1) and write for a cell Ω_{ij} in the interior,

$$\frac{\partial(\rho_{i,j}\theta_{i,j})}{\partial t} = -R_{i,j} \tag{14}$$

$$R_{i,j} = -O_{d,i,j} - f_{i,j} \tag{15}$$

$$i = 2, \ldots, n_{x1} - 1,\ j = 2, \ldots, n_{x2} - 1$$

For a cell $\Omega_{i,j}$ in the interior, the discrete spatial approximation is

$$\begin{aligned} O_{d,i,j} = \frac{1}{\Delta x_1} \Bigg[& \rho_{i+1/2,j} K_{e,i+1/2,j} \left(\frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x_1} - \rho_{i+1/2,j} g_{u1} \right) \\ & - \rho_{i-1/2,j} K_{e,i-1/2,j} \left(\frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta x_1} - \rho_{i-1/2,j} g_{u1} \right) \Bigg] \\ + \frac{1}{\Delta x_2} \Bigg[& \rho_{i,j+1/2} K_{e,i,j+1/2} \left(\frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta x_2} - \rho_{i,j+1/2} g_{u2} \right) \\ & - \rho_{i,j-1/2} K_{e,i,j-1/2} \left(\frac{\psi_{i,j} - \psi_{i,j-1}}{\Delta x_2} - \rho_{i,j-1/2} g_{u2} \right) \Bigg] \end{aligned} \tag{16}$$

where the subscripts in $\mathbf{g}_u = [g_{u1}, g_{u2}]^T$ indicate values for each coordinate, and quantities with a 1/2 subscript denote values estimated at cell interfaces.

For the interface values, we use a harmonic average for saturated hydraulic conductivity, the arithmetic average for density, and upwind the relative permeability

$$\rho_{i+1/2,j} = [\rho(\psi_{i+1,j}) + \rho(\psi_{i,j})]/2 \tag{17}$$

$$
\begin{aligned}
K_{e,i+1/2,j} &= k_{r,i+1/2,j} K_{s,i+1/2,j} \\
K_{s,i+1/2,j} &= 2K_{s,i,j} K_{s,i+1,j} / (K_{s,i,j} + K_{s,i+1,j}) \\
k_{r,i+1/2,j} &= \begin{cases} k_r(\psi_{i+1,j}) & \text{if } \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x_1} > \rho_{i+1/2,j} g_{u1} \\ k_r(\psi_{i,j}) & \text{otherwise} \end{cases}
\end{aligned} \tag{18}
$$

The corresponding terms along the other coordinate axis are defined symmetrically.

The physical boundaries are located at the nodes (cell centers), so specifying Dirichlet conditions in pressure or volume fraction is straightforward. For example, at a cell $\Omega_{i,j} \subset \Gamma_D$, we set

$$
C(\psi_{i,j} - \psi^b_{i,j}) = 0, \tag{19}
$$

where C is a scaling factor that gives the boundary equation roughly the same scaling as the interior nodes.

For cells along Γ_N, we use linear extrapolation to apply the flux at the exterior (artificial) cell boundary rather than the cell center. This approach allows us to use the same nodal equation at the boundary node as we do at the interior points for $\Omega_{i,j} \subset \Gamma_N$. If, for example, $\Omega_{1,j} \subset \Gamma_N$, we set

$$
u_{1,1/2,j} = -2u^b_{1,1,j} - u_{1,3/2,j} \tag{20}
$$

at the fictitious left cell boundary. Here $\mathbf{u} = [u_1, u_2]^T$.

3.3 Temporal approximation

Since we consider both direct solution of the steady-state problem eqn (7) and integration of eqn (1) to steady state, we begin by applying the cell-centered

finite difference spatial approximation to eqn (13) in a method of lines (MOL) context

$$A(\psi_{i,j})\frac{\partial \psi_{i,j}}{\partial t} = -R_{i,j} \qquad i = 2, \ldots, n_{x1} - 1,\ j = 2, \ldots, n_{x2} - 1 \tag{21}$$

with the Dirichlet and Neumann boundary conditions given by eqns (19) and (20).

The semi-discrete system corresponding to eqn (21) can be written as a set of differential algebraic equations (DAE's)

$$\mathbf{F}(t, \mathbf{y}, \mathbf{y}') = 0 \tag{22}$$

where $\mathbf{F}$ represents a set of equations that depend on time t, a set of dependent variables $\mathbf{y}$, and a set of first-order derivatives with respect to time of these dependent variables, $\mathbf{y}'$.

A variety of approaches can be used to integrate eqn (22). To illustrate the structure of the nonlinear system for transient problems, we use a backward Euler approximation to convert eqn (22) to a fully discrete system [13, 21].

$$\mathbf{G}(t^{n+1}, \mathbf{y}^{n+1}, \frac{1}{\Delta t^{n+1}}(\mathbf{y}^{n+1} - \mathbf{y}^{n})) = 0 \tag{23}$$

3.3.1 Nonlinear solution for the transient problem

At each time level, a full time integration approach such as backward Euler must solve the nonlinear system eqn (23). A general Newton iteration for eqn (23) can be written

$$\left[\mathbf{J}^{n+1,s_n}\right]\left\{\Delta \mathbf{y}^{s_n+1}\right\} = -\left\{\mathbf{G}^{n+1,s_n}\right\} \tag{24}$$

where $\{\Delta \mathbf{y}^{s_n+1}\} = \{\mathbf{y}^{n+1,s_n+1}\} - \{\mathbf{y}^{n+1,s_n}\}$ and s_n is a nonlinear iteration index. The Jacobian, $\mathbf{J}$, is formed by differentiating eqn (23) with respect to $\mathbf{y}$. For eqn (21), we can write $\mathbf{J}$ as

$$\left[\mathbf{J}^{n+1,s_n}\right] = \frac{1}{\Delta t^{n+1}}\left[\mathbf{A}^{n+1,s_n}\right] + \left[\frac{\partial(\mathbf{A}\mathbf{y}')}{\partial \mathbf{y}}^{n+1,s_n}\right] - \left[\frac{\partial \mathbf{O}_d}{\partial \mathbf{y}}^{n+1,s_n}\right] \tag{25}$$

where $\mathbf{A}$ is diagonal with $[\mathbf{A}]_{l,l} = A(\psi_{i,j})$, $\partial \mathbf{A}\mathbf{y}'/\partial \mathbf{y}$ is also diagonal, and $\partial \mathbf{O}_d/\partial \mathbf{y}$ will be banded with seven non-zero entries. Here, l is a global identifier corresponding to cell $\Omega_{i,j}$ with, for example, $l = (j-1)n_{x1}+i$ for $i = 1, \ldots, n_{x1}$, $j = 1, \ldots, n_{x2}$.

For unknowns along Γ_D, $\mathbf{J}$ is simply (see eqn (19))

$$\left[\mathbf{J}^{n+1,s_n}\right]_{l,l} = C \tag{26}$$

3.4 Ψtc approximation

Ψtc attempts to find a solution to eqn (7) by integrating eqn (21) to steady state. The approach is straightforward and can included in many existing steady-state or transient solvers with minor modifications. The fully discrete Ψtc system can be obtained from eqn (21) by first applying a backward Euler time discretization as in eqn (23) and then using Newton's method with $\mathbf{y}^n$ as the initial guess.

3.4.1 Solution for Ψtc update

While applying multiple iterations of Newton's method is possible, the form of Ψtc which we present here performs only a single Newton update for eqn

(23) [23]. The resulting equation for the Ψtc iterate is

$$[\mathbf{J}^n]\left\{\Delta\mathbf{y}^{n+1}\right\} = -\{\mathbf{R}^n\} = \{\mathbf{O}_d(\mathbf{y}^n)\} + \{\mathbf{f}^n\} \tag{27}$$

with $\{\Delta\mathbf{y}^{n+1}\} = \{\mathbf{y}^{n+1}\} - \{\mathbf{y}^n\}$ since only one Newton iteration is performed and $\{\mathbf{y}^{n+1,0}\} = \{\mathbf{y}^n\}$. $\mathbf{J}$ has the same form as eqn (25), but without a derivative of the accumulation term

$$[\mathbf{J}^n] = \frac{1}{\Delta t^{n+1}}[\mathbf{A}^n] - \left[\frac{\partial \mathbf{O}_d}{\partial \mathbf{y}}^n\right] \tag{28}$$

Note that when Δt is small enough $[\mathbf{J}^n] \approx \frac{1}{\Delta t^{n+1}}[\mathbf{A}^n]$ and therefore

$$\Delta\mathbf{y}^{n+1} \approx -\Delta t^{n+1}[\mathbf{A}^n]^{-1}\{\mathbf{R}^n\} \tag{29}$$

which is the update corresponding to the forward Euler method applied to the ODE form of RE.

3.4.2 Ψtc step selection

Ψtc solves a series of problems of the form in eqn (27), while adapting the time step Δt^{n+1} based on the intermediate solution's behavior. There are a number of common strategies, for selecting Δt^{n+1}, including the switched evolution relaxation (SER) [23, 15, 11]

$$\Delta t^{n+1} = \Delta t^n \frac{\|\mathbf{R}^{n-1}\|}{\|\mathbf{R}^n\|} \tag{30}$$

The temporal truncation error approach (TTE) attempts to control the time step based on the local temporal truncation error [23, 15, 11]. It chooses Δt^{n+1}

so that

$$\left| \frac{(\Delta t^{n+1})^2}{2(1+|y_l^n|)} \frac{\partial^2 y_l}{\partial t^2}(t^n) \right| \leq \tau \tag{31}$$

for each component of the solution y_l and some constant τ. Here, we estimate $\partial^2 y_l / \partial t^2$ at t^n by [11]

$$\frac{\partial^2 y_l}{\partial t^2}(t^n) \approx \frac{2}{\Delta t^n + \Delta t^{n-1}} \left[\frac{y_l^n - y_l^{n-1}}{\Delta t^n} - \frac{y_l^{n-1} - y_l^{n-2}}{\Delta t^{n-1}} \right] \tag{32}$$

For both SER and TTE, we also enforce an upper bound Δt_{max} on the chosen time step.

3.5 Newton's method for the steady-state problem

Following eqn (21), we can write a cell-centered finite difference spatial approximation of eqn (7) to solve the steady-state problem directly

$$\begin{aligned} R_{i,j} &= 0 \\ i &= 2, \ldots, n_{x1} - 1, \; j = 2, \ldots, n_{x2} - 1 \end{aligned} \tag{33}$$

with Dirichlet and Neumann boundary conditions given by eqns (19) and (20) without temporal dependence.

The Newton update for eqn (33) is

$$\begin{aligned} [\mathbf{J}^{s_n}] \left\{ \Delta \mathbf{y}^{s_n+1} \right\} &= - \left\{ \mathbf{R}^{s_n} \right\} \\ [\mathbf{J}] &= - \left[\frac{\partial \mathbf{O}_d}{\partial \mathbf{y}} \right] \end{aligned} \tag{34}$$

3.5.1 Globalization techniques

A number of globalization strategies are commonly used to address the sensitivity of Newton's method to initial guess. The Armijo line-search technique scales the original Newton update by a factor chosen to take a step as close to the original update as possible while insuring a sufficient decrease in the nonlinear residual [18]. At iteration level s_n for eqn (34), the Armijo line search can be written

(1) Solve eqn (34) for $\Delta\mathbf{y}^{s_n+1}$, and set $\lambda = \lambda_+ = 1$

(2) While $\|\mathbf{R}(\mathbf{y}^{s_n} + \lambda\Delta\mathbf{y}^{s_n+1})\| > (1 - \epsilon_s\lambda)\|\mathbf{R}(\mathbf{y}^{s_n})\|$

Choose λ_+

if $\lambda_+ < \sigma_{min}\lambda$, $\lambda_+ = \sigma_{min}\lambda$

else if $\lambda_+ > \sigma_{max}\lambda$, $\lambda_+ = \sigma_{max}\lambda$

$\lambda = \lambda_+$

(3) Set $\{\mathbf{y}^{s_n+1}\} = \{\mathbf{y}^{s_n}\} + \lambda\{\Delta\mathbf{y}^{s_n+1}\}$

where ϵ_s is a parameter controlling the amount of decrease required in the nonlinear residual and λ is the final scaling factor. Here, we chose λ_+ so that it minimized a three-point parabolic approximation of $\|R(\mathbf{y}^{s_n} + \lambda\Delta\mathbf{y}^{s_n+1})\| = f(\lambda)$. Bounds on λ_+ are dictated by σ_{min} and σ_{max}. The details of this approach can be found in Kelley [22]. For this work we set $\sigma_{min} = 0.1$,$\sigma_{max} = 0.55$, and $\epsilon_s = 10^{-4}$. The line search can fail if λ becomes too small. In this case, the nonlinear solver is said to have failed due to line-search stagnation [11].

To avoid evaluation errors in the constitutive relations and to increase the robustness of the iterations, we also include a test to make sure that the solution is within broad, physically relevant bounds.

(1) Solve eqn (34) for $\Delta \mathbf{y}^{s_n+1}$, and set $\lambda = 1$

(2) While $y_l^{s_n} + \lambda \Delta y_l^{s_n+1} \notin [y_{min}, y_{max}] \quad \forall\, l$

$$\lambda = \lambda/2$$

(3) Set $\{\mathbf{y}^{s_n+1}\} = \{\mathbf{y}^{s_n}\} + \lambda \{\Delta \mathbf{y}^{s_n+1}\}$

In the numerical experiments presented below, we use an interval $y_{min} = -100\ [m]$, $y_{max} = 100\ [m]$.

3.6 Picard iteration

Picard iteration has been used widely in the solution of RE [10, 19]. In brief, a Picard linearization of eqn (33) can be formulated as the Newton update in eqn (34) but with an approximate Jacobian in which the coefficient derivatives $\partial k_r/\partial \psi$ and $\partial \rho/\partial \psi$ are omitted from $\partial \mathbf{O}_d/\partial \mathbf{y}$ [29, 25]. The performance of Picard iteration and Newton approaches have been compared in several works [29, 25, 27].

In many cases, Newton's method with line search has proven more robust than a straightforward application of Picard iteration [27]. However, a hybrid Newton-Picard algorithm has also been suggested for reducing the sensitivity of Newton's method to the quality of the initial guess [29, 8]. This approach performs an initial number of iterations for eqn (34) using the Picard approximation for $\mathbf{J}$ and then switches to a Newton update with the full Jacobian [29, 8]. The motivation for this approach is that the Picard iterations are, in general, cheaper than their Newton counterparts.

Ideally, the switch to Newton's method is chosen so that (1) a minimal number

of Picard iterations are performed; and, (2) upon switching, $\mathbf{y}^{s_n}$ is sufficiently close to the true to solution that the asymptotic convergence rate for the Newton updates is realized. There are several possible criteria for determining the crossover from Picard to Newton iteration, including $\|\Delta\mathbf{y}^{s_n}\| < \epsilon_c$ [8]. Alternatively, one can base the switch on sufficient decrease in the initial residual,

$$\|\mathbf{R}^{s_n+1}\| < \epsilon_c \|\mathbf{R}^0\| \tag{35}$$

We use eqn (35) in the numerical results presented below.

3.7 Linear system solution

We test five algorithms for solving the linear systems arising in the nonlinear iteration. As a direct solver, we use both the banded LU decomposition from LAPACK (LU-B) [1], as well as the implementation of LU from the SPOOLES package (LU-S) [2, 3] in PETSc [6, 7, 5]. We also test the iterative method BiCGstab using ILU preconditioning with zero fill from PETSc (BiCGstab-ILU), and a two-level hybrid additive Schwarz domain decomposition method (BiCGstab-HAS) [17].

4 Results

In the following sections we will present results of several numerical experiments and compare the behavior of Ψtc with Newton's method and a hybrid Newton-Picard iteration. First we summarize the test problems on which the numerical experiments were carried out.

4.1 Test problems

4.1.1 Problem 1: infiltration example

The first test problem is a relatively simple one-dimensional example and provides a case where each of the methods should perform well. We consider each test case as a transient problem with a corresponding steady-state solution. The first example simulates infiltration in vertical domain $\Omega = [0, X_1]$. Initial conditions for the infiltration are set to static equilibrium with the water table, located at the bottom of the domain. Constant Dirichlet boundary conditions are set at the top so that the steady-state solution contains both saturated and unsaturated regions. Table 1 summarizes the relevant physical parameters and auxiliary conditions.

4.1.2 Problem 2: hillslope example

The spatial domain for the second problem, $\Omega = [0, X_1] \times [0, X_2]$, is illustrated in Figure 1. The temporal domain is $t \in [0, T]$ with T chosen so that the solution is at steady state. The relevant physical parameters and auxiliary conditions for Problem 2 are given in Tables 2-4. The log of the saturated hydraulic conductivity is given in Figure 2. Note that the domain is rotated 45 degrees to simulate a simple hillslope. The domain has block-heterogeneous medium properties ranging from clay to sand.

The boundary and initial conditions are configured to reflect an impermeable bedrock at the X_2^- boundary, no flow at the X_1^+ boundary, and static, saturated equilibrium along X_1^-. The surface boundary condition at X_2^+ is a nonlinear flux boundary condition that simulates infiltrating precipitation

Table 1

Fluid and domain properties for Problem 1

Variable	Value	Units
g_{u1}	-1.0	$[-]$
$\|\mathbf{g}\|$	7.321×10^{10}	$[m/d^2]$
ϱ_0	998.2	$[kg/m^3]$
β	6.564×10^{-20}	$[m\cdot d/kg]$
p_0	0	$[kg/m\cdot d]$
X_1	10	$[m]$
$\psi^b(x_1=0)$	0	$[m]$
$\psi^b(x_1=10)$	-0.05	$[m]$
$\psi^0(x_1)$	$-x_1$	$[m]$
n_v	4.264	$[-]$
α_v	5.470	$[m^{-1}]$
K_s	5.040	$[m/d]$

until the surface becomes saturated. After saturated conditions are reached at the surface, the outward normal flux increases to zero and becomes positive as the pressure rises above atmospheric pressure. This condition reflects a well-drained surface that permits very little ponding at the surface. The boundary conditions can be summarized as follows.

$$\psi_{X_1^-} = x_2 \tag{36}$$

$$u^b_{X_1^+} = 0 \tag{37}$$

$$u^b_{X_2^-} = 0 \tag{38}$$

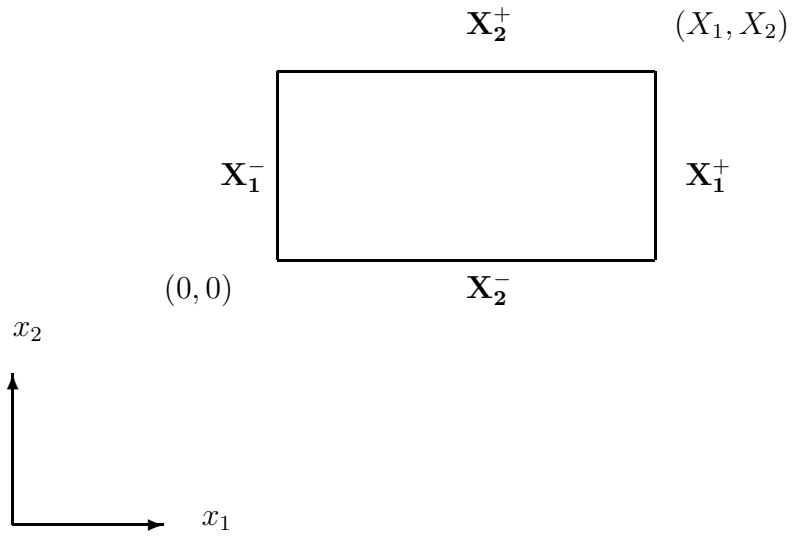
$$u^b_{X_2^+} = \left\{ \begin{array}{c} u^r, \quad \psi < 0 \\ u^r + K_s^s \psi, \; \psi > 0 \end{array} \right\} \tag{39}$$

where u^r is the precipitation rate and K_s^s is the saturated conductivity at the

surface. For the numerical experiments presented, $K_s^s = 10\ [m/d]$

The steady-state solution of the volume fraction is given in Figure 3. This solution was computed by integrating the transient problem to $T = 1700\ [d]$ at which time $\|R\|_\infty < 10^{-5}$. The solution was obtained using a DAE integrator with relative and absolute integration tolerances of 10^{-4} [20, 21]. The transient solution exhibits steep moving fronts throughout the domain while the equilibrium solution shown maintains a number of steep moisture gradients due to the layering of the media.

Fig. 1. Problem domain Ω



4.2 Numerical experiments

For both test problems, we performed several numerical experiments on a series of grids. On each grid, we ran Newton's method with a quadratic line search (NILS), the Newton-Picard hybrid approach (PIH), as well as two Ψtc

Table 2

Media distribution for Problem 2

Medium Type	P1	P2	P1	P1	P1	P1	P1	P2	P2	P2
Left $[m]$	0	0	0.7	1.4	2.9	7.1	8.6	2.9	6.4	9.3
Bottom $[m]$	0	0.1	0.1	0.1	0.2	0.1	0.3	0.1	0.2	0.1
Right $[m]$	10	0.71	1.4	2.9	6.4	9.3	9.3	7.1	7.1	10
Top $[m]$	0.1	1	0.8	0.3	0.3	0.3	0.8	0.2	0.3	1
Medium Type	P2	P2	P2	P2	P2	P3	P3	P3	P4	P1
Left $[m]$	3.6	1.4	0.7	1.4	7.1	1.4	2.9	5.0	1.4	5.0
Bottom $[m]$	0.4	0.6	0.8	0.7	0.7	0.4	0.7	0.4	0.3	0.5
Right $[m]$	5.0	8.6	9.3	2.9	8.6	3.6	7.1	8.6	8.6	8.6
Top $[m]$	0.6	0.7	1	0.8	0.8	0.6	0.8	0.5	0.4	0.6

Table 3

Media properties for Problem 2

Medium Type	θ_s	θ_r	n_v	α_v $[m^{-1}]$	K_s $[m/d]$
P1	0.41	0.07749	2.090	0.244	1.10808×10^{-5}
P2	0.40	0.03120	4.264	5.470	5.04000×10^{0}
P3	0.39	0.03822	2.370	0.478	1.80100×10^{-3}
P4	0.39	0.02691	3.264	0.244	4.04000×10^{0}

methods, SER and TTE, until $\|R\|_2 < (\|R_0\|_2 + 1)10^{-5}$. The initial guess was taken to be the initial conditions for the corresponding transient problem. For the NILS and PIH calculations, we enforced a maximum number of nonlinear iterations $s_n \leq N_{max} = 1000$, while the maximum number of steps for the SER and TTE runs was $n \leq N_{max} = 5000$. The maximum number of line searches allowed was 1000 and the sufficient decrease parameter for Newton line search was $\epsilon_s = 10^{-4}$. The PIH approach used a value of $\epsilon_c = 10^{-2}$ in its switching strategy, eqn (35). The Ψtc methods used an initial time step

Table 4

Fluid and domain properties for Problem 2

Variable	Value	Units
g_{u1}	-0.7071	$[-]$
g_{u1}	-0.7071	$[-]$
$\|\mathbf{g}\|$	7.321×10^{10}	$[m/d^2]$
ϱ_0	998.2	$[kg/m^3]$
β	6.564×10^{-20}	$[m \cdot d/kg]$
p_0	0	$[kg/m \cdot d]$
X_1	10	$[m]$
X_2	1	$[m]$
u^r	-0.4	$[m/d]$

Fig. 2. $\log(K_s)$ for Problem 2

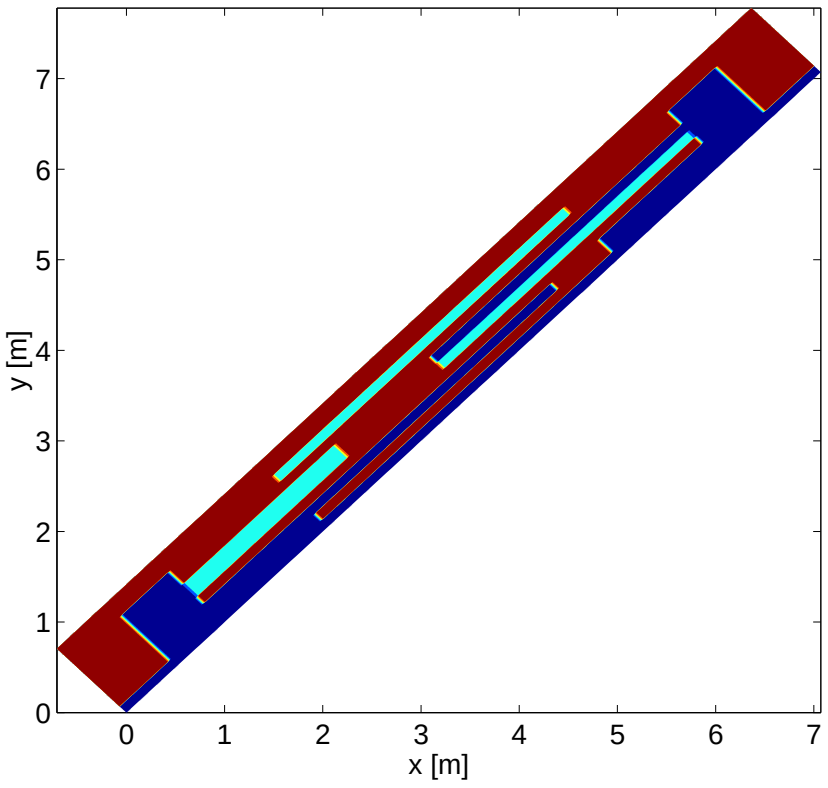
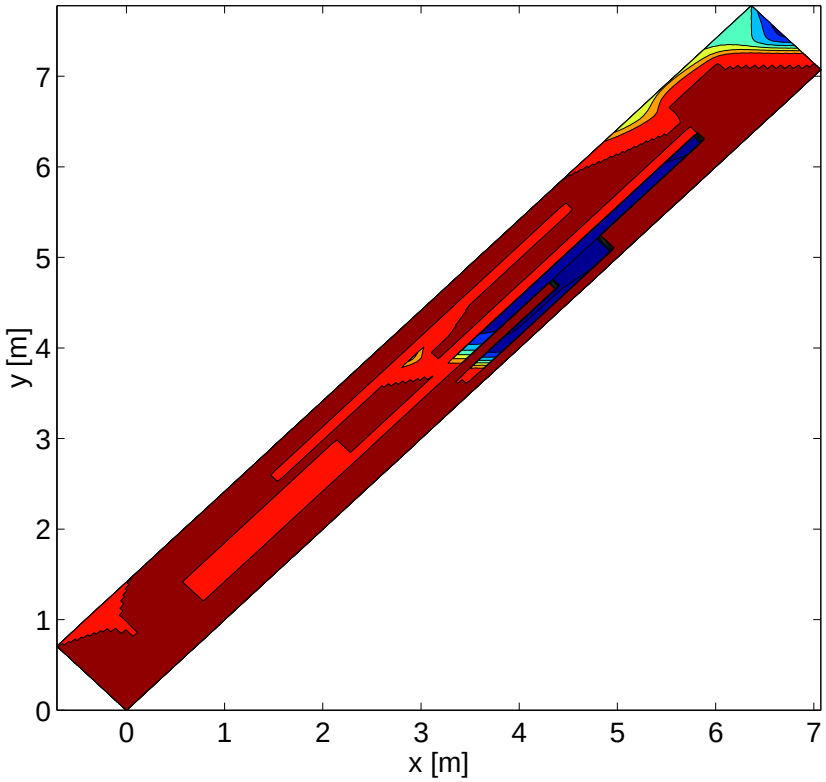


Fig. 3. Steady-state volume fraction for second test problem



of $\Delta t^0 = 4.050 \times 10^{-5}$, a maximum time step $\Delta t_{max} = 10^5$, and the TTE algorithm used $\tau = 1.0$. The relevant parameters are summarized in Table 5.

The linear systems for Problem 1 were solved using the LU-B decomposition from LAPACK with each method. For Problem 2, we ran the numerical experiments for each of the four steady-state solution methods with the LU-S direct solver as well the two iterative methods (BiCGstab-ILU and BiCGstab-HAS). A relative residual test was used for BiCGstab with a tolerance of 10^{-6}. The maximum number of linear iterations allowed was 2000 except where noted. The simulations were performed on a Pentium 4 (2.53 Ghz) workstation with 256 Mbytes of RAM running Redhat Linux 7.3. The elapsed time for the simulations was recorded using the ANSI C `clock` and `time` intrinsics.

The results for Problem 1 on spatial grids of size n_e =41–161 are presented in

Table 5

Numerical methods summary

	NILS	PIH
N_{max}	1000	1000
ϵ_s	10^{-4}	-
ϵ_c	-	10^{-2}
	TTE	SER
N_{max}	5000	5000
Δt^0	4.050×10^{-5}	4.050×10^{-5}
Δt_{max}	10^5	10^5
τ	1.0	-

Table 6. The labels are as follows:

- Jeval-Jacobian evaluation
- Feval-function evaluation
- Steps-attempted iterations for method
- NLI-nonlinear iteration
- NLF-nonlinear iteration failure
- LI-linear iteration
- LF-linear iteration failure
- Clock-wall clock time (seconds)
- X-failed to converge

The simulations for Problem 1 were small and involved a homogeneous medium. As a result, we expected the methods to have little difficulty in its solution. Both the NILS method and the PIH iteration performed well, even though the NILS method employed a number of line searches on each spatial grid.

Table 6

LU-B runs for Problem 1

	Jeval	Feval	Steps	NLI	NLF	LI	LF	LS	Clock
				$n_e = 41$					
NILS	30	299	30	30	0	0	0	102	0.05
PIH	14	33	14	14	0	0	0	1	0.01
TTE	51	103	51	0	0	0	0	0	0.06
SER	83	167	83	0	0	0	0	0	0.04
				$n_e = 81$					
NILS	26	253	26	26	0	0	0	88	0.04
PIH	15	35	15	15	0	0	0	1	0.03
TTE	104	209	104	0	0	0	0	0	0.14
SER	168	337	168	0	0	0	0	0	0.09
				$n_e = 161$					
NILS	25	329	25	25	0	0	0	128	0.08
PIH	16	35	16	16	0	0	0	0	0.06
TTE	207	410	202	0	0	0	0	0	1.43
SER	353	705	351	0	0	0	0	0	0.33

PIH took roughly half as many steps as NILS and needed only one line search once it switched to the Newton iteration, indicating that the initial Picard iterations were more successful in advancing the intermediate iterates closer to the steady-state solution than Newton's method with a line search alone. The Ψtc methods also converged to the correct solution, but required significantly more steps than either the NILS or PIH approaches. To illustrate the methods' performance, Figure 4 shows the residual history for NILS and PIH. Both methods achieved quadratic convergence as they approached the root.

Problem 2 was more challenging than Problem 1 due to its dimensionality, heterogeneous medium, and the nonlinear boundary condition at the surface. Results of the numerical experiments with LU-S and BiCGstab are presented in Tables 7-9. The increased wall clock times, iteration counts, and line searches reflect the added difficulty of Problem 2. NILS converged for every linear solver and spatial grid in Tables 7-9. It required a similar number of nonlinear iterations and line searches for a particular grid, regardless of the linear solver used. Unlike NILS, PIH failed for the cases considered. With one exception where it failed in the initial linear solve, PIH did not reduce the original nonlinear residual sufficiently to satisfy eqn (35) and switch to NILS. In these cases, it exhausted the allowed number of nonlinear iterations.

Both the Ψtc methods converged for each spatial grid and linear solver combination in Tables 7-9, with TTE consistently between 2 and 5 times faster than SER. The run times for NILS and TTE were similar for most of the simulations. NILS was more efficient with the LU-S solver, particularly for the 81×81 grid. On the other hand, TTE was more efficient for the BiCGstab calculations. The difference in run times increased for the HAS preconditioner, where TTE was 3 and 1.4 times faster than NILS on the 41×41 and 81×81 grids respectively.

The results also indicate a difference in the way NILS and TTE behaved as the spatial grid was refined. Namely, NILS required roughly the same number of iterations to converge regardless of spatial grid or linear solver, while the number of steps taken by TTE grew noticeably as n_e increased. To investigate the performance of TTE, we ran an additional set of calculations where τ was set to 10^{-3} and scaled by n_e for each grid. The corresponding τ values ranged from 0.121 on the 11×11 grid to 25.9 on a grid with $n_e = 161 \times 161$. As a

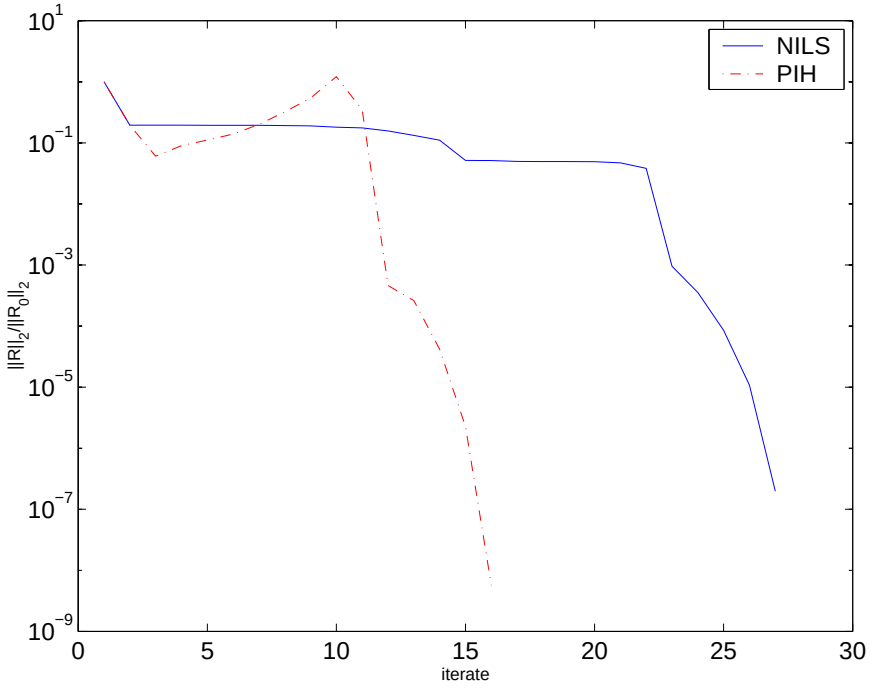
result, the TTE time step selection was slightly more conservative on coarser grids than the original choice of $\tau = 1$ and was more aggressive on the finer grids. Table 10 contains the results for TTE with LU-S and BiCGstab for $n_e = 11 \times 11$ to $n_e = 161 \times 161$ as well as the results for NILS on the finest grid.

With the use of a scaled τ, Steps for TTE was roughly the same for $n_e = 11 \times 11$ to $n_e = 81 \times 81$ and each linear solver. There was still, however, an increase in the number of steps for the finest grid, particularly for the BiCGstab-ILU solver. TTE with the scaled τ took slightly longer than the original $\tau = 1$ calculations on the coarser grids, but reduced the total simulation time as the grids were refined. There was a corresponding improvement for TTE on the larger spatial grids with all three solvers. The computational effort for NILS and TTE with τ scaled was essentially the same for LU-S except for the finest grid, where NILS was 1.6 times faster. For BiCGstab-ILU and BiCGstab-HAS, TTE with τ scaled was twice as fast as the simulations with $\tau = 1$ on the $n_e = 81 \times 81$ grid.

We also note that BiCGstab-ILU did not perform well with the $n_e = 161 \times 161$ system. Simulations with each of the steady-state solution methods and BiCGstab-ILU typically required significantly less computational effort than BiCGstab-HAS for the coarser grids. However, for $n_e = 161 \times 161$ the total number of steps, linear iterations, and simulation time increased significantly for TTE while NILS failed after both 2000 and 10000 linear iterations.

As an example of the progress of the Ψtc iterations in comparison to the Newton and Newton-Picard iterations, we graphed the history of $\|R\|_2$ versus iteration for SER on the $n_e = 81 \times 81$ grid with the LU-S linear solver in Figure

Fig. 4. NILS/PIH residual history, Problem 1



5. As it neared the steady-state solution, the method's convergence steepened. This reflects the convergence of the Newton iteration that Ψtc reduces to near the root. Unlike the globalized Newton iteration, however, Ψtc does not enforce a decreasing sequence of residuals, and Figure 5 shows increased residuals at some points in the SER calculations. The iteration history of TTE was more extreme, often oscillating wildly before nearing the steady-state solution. Figure 6 shows the TTE calculation with $\tau = 1$ on the $n_e = 81 \times 81$ grid. For some problems, it has been found that performing multiple sub-iterations for eqn (27) can improve convergence [23, 15]. However, we investigated this alternative for the SER and TTE updates for the second test problem and found no improvement for either approach.

Table 7

LU-S runs for Problem 2

	Jeval	Feval	Steps	NLI	NLF	LI	LF	LS	Clock
				$n_e = 11 \times 11$					
NILS	83	3901	83	83	0	83	0	1806	1.5
PIH	1000	6173	1000	1000	1	1000	0	0	X
TTE	49	99	49	0	0	49	0	0	0.26
SER	145	291	145	0	0	145	0	0	1.65
				$n_e = 21 \times 21$					
NILS	132	7057	132	132	0	132	0	3291	2.55
PIH	1000	2239	1000	1000	1	1000	0	0	X
TTE	91	183	91	0	0	91	0	0	2.59
SER	379	759	379	0	0	379	0	0	7.5
				$n_e = 41 \times 41$					
NILS	106	4725	106	106	0	106	0	2181	9.92
PIH	1000	2885	1000	1000	1	1000	0	0	X
TTE	173	346	172	0	0	173	0	0	12.23
SER	866	1732	865	0	0	866	0	0	53.29
				$n_e = 81 \times 81$					
NILS	149	7369	149	149	0	149	0	3412	56.44
PIH	1000	3449	1000	1000	1	1000	0	0	X
TTE	450	888	437	0	0	450	0	0	164.43
SER	2068	4135	2066	0	0	2068	0	0	709.67

5 Discussion

We began with the goal of identifying effective methods for the steady-state solution of RE. To this end, we performed several numerical experiments for

Table 8

BiCGstab-ILU runs for Problem 2

	Jeval	Feval	Steps	NLI	NLF	LI	LF	LS	Clock
				$n_e = 11 \times 11$					
NILS	83	3901	83	83	0	1297	0	1806	1.58
PIH	1000	6173	1000	1000	1	7371	0	0	X
TTE	49	99	49	0	0	210	0	0	0.3
SER	145	291	145	0	0	445	0	0	1.77
				$n_e = 21 \times 21$					
NILS	134	7205	134	134	0	6895	0	3360	4.5
PIH	1000	2239	1000	1000	1	19419	0	0	X
TTE	98	197	98	0	0	970	0	0	2.08
SER	378	757	378	0	0	2559	0	0	7.14
				$n_e = 41 \times 41$					
NILS	111	4863	111	111	0	11153	0	2239	17.01
PIH	1000	2893	1000	1000	1	38981	0	0	X
TTE	177	354	176	0	0	2215	0	0	13.49
SER	865	1730	864	0	0	11693	0	0	54.44
				$n_e = 81 \times 81$					
NILS	145	7127	145	145	0	31799	0	3301	148.57
PIH	1	3	1	1	0	2000	1	0	X
TTE	424	847	422	0	0	7934	0	0	136.65
SER	2068	4135	2066	0	0	50243	0	0	604

Newton's method with two globalization techniques (NILS and PIH) as well as two versions of Ψtc (SER and TTE). Various observations can be made based on the results, which reflect a range of difficulty for the one and two-

Table 9

BiCGstab-HAS runs for Problem 2

	Jeval	Feval	Steps	NLI	NLF	LI	LF	LS	Clock
				$n_e = 11 \times 11$					
NILS	84	3959	84	84	0	4200	0	1833	1.13
PIH	1000	6173	1000	1000	1	24663	0	0	X
TTE	49	99	49	0	0	425	0	0	0.38
SER	146	293	146	0	0	1076	0	0	1.99
				$n_e = 21 \times 21$					
NILS	131	7059	131	131	0	46899	0	3293	26.3
PIH	1000	2239	1000	1000	1	51097	0	0	X
TTE	88	177	88	0	0	1628	0	0	2.63
SER	379	759	379	0	0	5799	0	0	9.04
				$n_e = 41 \times 41$					
NILS	110	4991	110	110	0	24471	0	2305	69.02
PIH	1000	3019	1000	1000	1	74767	0	0	X
TTE	168	336	167	0	0	4814	0	0	23.93
SER	866	1732	865	0	0	26241	0	0	117.69
				$n_e = 81 \times 81$					
NILS	148	7315	148	148	0	30762	0	3389	395.15
PIH	1000	3449	1000	1000	1	69639	0	0	X
TTE	439	867	427	0	0	11964	0	0	288.42
SER	2068	4135	2066	0	0	78706	0	0	1377.61

dimensional variably saturated flow problems examined in this work.

The use of a line search or hybrid Picard iteration made Newton's method more robust for Problem 1. PIH required roughly half the iterations of NILS,

Table 10

Runs with τ scaled by n_e for Problem 2

	n_e	Jeval	Feval	Steps	NLI	NLF	LI	LF	LS	Clock
					LU-S					
TTE	11×11	114	229	114	0	0	114	0	0	1.57
TTE	21×21	127	255	127	0	0	127	0	0	2.96
TTE	41×41	125	250	124	0	0	125	0	0	8.16
TTE	81×81	146	281	134	0	0	146	0	0	58
TTE	161×161	248	458	209	0	0	248	0	0	667.07
NILS	161×161	232	12961	232	232	0	232	0	6043	429.58
					BiCGstab-ILU					
TTE	11×11	113	227	113	0	0	447	0	0	0.69
TTE	21×21	129	259	129	0	0	1132	0	0	2.36
TTE	41×41	130	260	129	0	0	1739	0	0	9.72
TTE	81×81	148	289	140	0	0	3474	0	0	65.85
TTE	161×161	806	1185	378	0	0	92232	0	0	2330
NILS	161×161	1	3	1	1	0	10000	1	0	X
					BiCGstab-HAS					
TTE	11×11	115	231	115	0	0	979	0	0	1.86
TTE	21×21	125	251	125	0	0	2173	0	0	4.28
TTE	41×41	131	262	130	0	0	3782	0	0	18.17
TTE	81×81	152	288	135	0	0	7192	0	0	131.9
TTE	161×161	283	472	188	0	0	20806	0	0	1386.07
NILS	161×161	236	13153	236	236	0	70484	0	6130	3397

Fig. 5. SER residual history, Problem 2

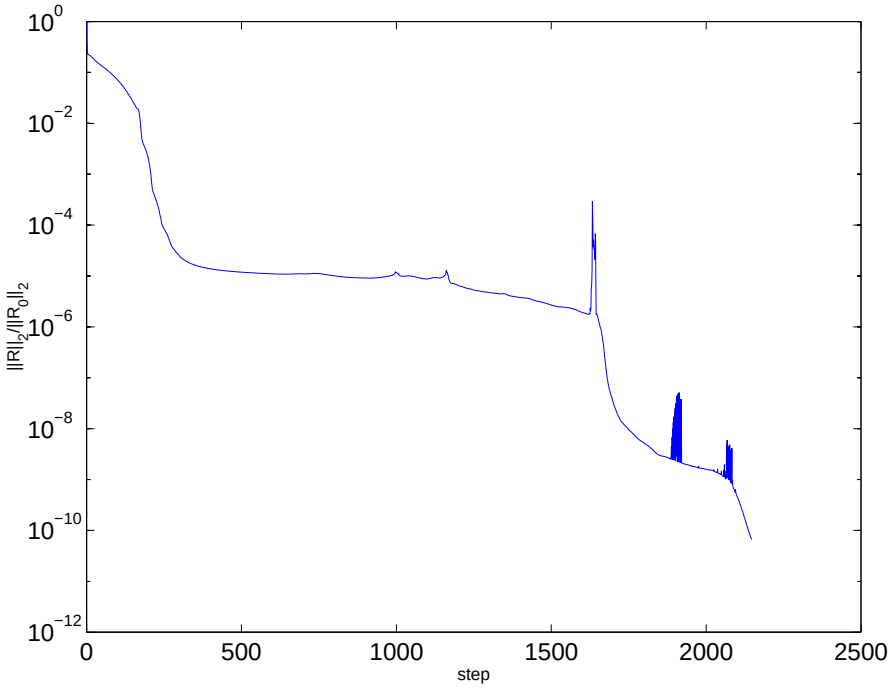
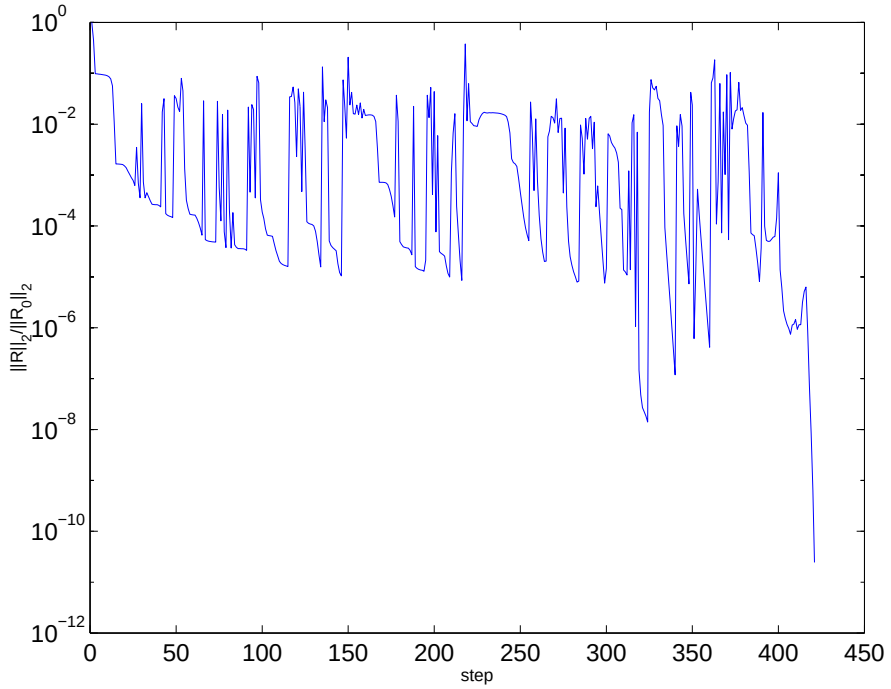


Fig. 6. TTE residual history, Problem 2



while both it and NILS took considerably fewer steps than either SER or TTE to reach steady state. The first test problem was intended to represent behavior for mild problems since the boundary conditions were simple and

the conductivity was homogeneous. The resulting linear systems were also small and tridiagonal. Under these ideal conditions, NILS had little difficulty, and PIH was able to speed convergence to the steady-state solution. The Ψtc methods offered no benefit in this situation.

The methods' relative performance changed though, as we moved to more realistic conditions. The controlling factor in their performance for Problem 2 was the difficulty associated with solving the resulting linear systems for each method. In addition to nonlinear boundary conditions, Problem 2 involved a two-dimensional, block-heterogeneous domain with medium properties corresponding to sand and clay. As a result, the linear systems for Problem 2 were more poorly scaled than in Problem 1. On the first step of the 81×81 grid the condition number of the NILS Jacobian was approximately 10^{19} and after 150 iterations was 10^{9} (condition numbers calculated using the `dgbtrf/dgbcon` routines from LAPACK).

Poor scaling of the Jacobians manifested itself in a number of ways, including the added work required by BiCGstab to converge with both preconditioners. In general, the difficulty of solving the linear systems can also translate into less accurate search directions as, for example, one is forced to use coarser linear solver tolerances to obtain results given computational limitations. The performance of NILS with the LU-S solver illustrates the impact of the Jacobian scaling on direct methods and the result of inaccurate search directions. The results in Table 7 indicate that NILS was the most efficient approach when LU-S was used for the LU decompositions. However, when we performed the tests with the LU solver from PETSc which does not include pivoting, the solution was less accurate, and NILS either exhausted the allowed nonlinear iterations or stagnated in the line search.

The PIH approach did not perform as it was intended for Problem 2 because the initial Picard iterations did not adequately reduce the initial residual. The convergence of PIH could have been improved by relaxing the switching criterion in eqn (35) or enforcing a maximum number of Picard iterations, so that it would have switched to the Newton updates before exhausting the allowed iterations. However, this would be, in essence, just NILS and would have masked the ineffectiveness of the initial Picard iterations for Problem 2.

In contrast, SER was robust but relatively inefficient. It converged for each of the runs and linear solvers, but was significantly slower than TTE for most cases. The two most efficient methods for Problem 2 were NILS and TTE. Their relative efficiency was dictated largely by the computational expense associated with the linear solvers. With both methods, the number of steps required for a given spatial grid was similar using the LU-S and BiCGstab solvers. However, the computational effort required changed significantly. Using a robust direct solver, NILS was three time faster than TTE with $\tau = 1$ on the 81×81 grid. However, it required four times as many total linear iterations with BiCGstab-ILU and three times as many using BiCGstab-HAS. As a result, it was 9% slower using BiCGstab-ILU and 28% slower using BiCGstab-HAS.

The total number of iterations for NILS was fairly consistent on the spatial grids considered. On the other hand, TTE with $\tau = 1$ did not scale as well when the spatial grids were refined. The total number of steps required to converge increased with n_e, making TTE more expensive. Varying τ based on n_e led to more consistent results for TTE and improved its performance for finer grids. NILS was still more efficient with a direct solver on the 161×161 grid. On the other hand, TTE was between 2 and 3 times faster than NILS on

the finer grids using the iterative linear solvers and converged for BiCGstab-ILU on the 161×161 grid when NILS failed.

The value of a number of parameters like $\epsilon_c, \epsilon_s, \sigma_{min}, \sigma_{max}$ and the linear solver tolerances for BiCGstab effected each of the methods considered to some degree. The performance of TTE for different τ values demonstrates that the the impact of parameter choices can be significant at times. Finding a successful configuration for a given method and problem usually requires some effort. While different choices of parameter values may lead to improved performance, the basic behavior of the methods was the same for the range of parameters we investigated. NILS was preferable for the simulations with a robust direct solver like LU-S, but BiCGstab performed better with the Ψtc algorithms. TTE was more aggressive than SER and more efficient than NILS when BiCGstab was used with either ILU or HAS preconditioning.

As a reference point, we also compared the performance of the Ψtc methods to time-accurate integration using a variable-order, variable step-size DAE integrator [20]. As might be expected, the full time integration was the most reliable approach for obtaining steady-state solutions, but the computational expense incurred was from 5 to 25 times greater than that of SER, which was similarly robust.

An initial guess closer to the final solution would have improved the performance of each of the methods considered. Since the goal of PIH and the Ψtc methods is to approximate a Newton iteration near the steady-state solution, one can expect the advantage of NILS to increase as the initial guess better approximates the steady-state solution. However, we used static equilibrium as a reasonable initial guess, since we are interested in evaluating the meth-

ods for problems where good initial guesses for the steady-state solution are difficult to obtain.

The range of spatial grids presented for the numerical experiments was relatively coarse and the resulting linear systems were small to moderate in size. Still, the Jacobians were ill-conditioned for Problem 2, which proved to be a significant test for the direct and iterative linear solvers. For larger systems arising from realistic two and three-dimensional problems, we can only expect the difficulties associated with the linear systems to become more severe. Moreover, large simulations will often have to be solved in parallel to obtain results in a reasonable timeframe. For the numerical experiments presented here, NILS combined with the LU-S solver was the most efficient approach on each spatial grid. However, preconditioned Krylov methods and sparse direct solvers have their own advantages and disadvantages depending on the problem and architecture [12, 16]. A general comparison of their relative merits is beyond the scope of this paper.

6 Conclusions

Our numerical experiments for Ψtc approaches as well as Newton's method with various globalization techniques lead us to the following conclusions and recommendations:

- For problems where use of a robust direct solver is feasible, Newton's method with a line search is the most efficient approach for obtaining steady-state solutions to RE.
- Using an initial number of Picard iterations for Newton's method with line

search can improve performance in some instances, but it does not necessarily lead to more robust performance for difficult problems.

- Inexact Newton methods with standard globalization techniques have particular difficulty when the Jacobian is poorly scaled due to factors such as heterogeneous conductivity fields.
- If Newton's method fails or performs poorly for a given steady-state problem, it is worth examining a range of linear solver and line-search parameters before abandoning a Newton approach.
- Ψtc is a relatively simple approach that can improve the efficiency and robustness of existing steady-state solvers for RE on difficult problems, particularly if iterative linear solvers are used.

Acknowledgments

The authors would like to thank Dr. T.-C. Yeh for several helpful suggestions.

The research of CEK and CTK was supported in part by National Science Foundation grants DMS-0070641, DMS-0112542, and Army Research Office grant DAAD19-02-1-0391. The efforts of MWF and CTM were supported by grants 5 P42 ES05948 from the National Institute of Environmental Health Sciences and DMS-0112653 from the National Science Foundation. Computational support was provided by the North Carolina Supercomputer Center.

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and

D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, 1992.

[2] C. Ashcraft and R. Grimes. Spooles: An object-oriented sparse matrix library. In *Proceedings of the 1999 SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, 1999. March 22-24, San Antonio.

[3] C. Ashcraft and R. Grimes. SPOOLES homepage. Technical Report http://www.netlib.org/linalg/spooles/spooles.2.2.html, 1999.

[4] R.G. Baca, J.N. Chung, and D.J. Mulla. Mixed transform finite element method for solving the non-linear equation for flow in variably saturated porous media. *International journal for numerical methods in fluids*, 24: 441–455, 1997.

[5] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser, Boston, MA, 1997.

[6] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. PETSc homepage. Technical Report http://www.mcs.anl.gov/petsc, Argonne National Laboratory, 1998.

[7] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. PETSc 2.0 users manual. Technical Report ANL-95/11- Revision 2.0.28, Argonne National Laboratory, 2000.

[8] L. Bergamaschi and M. Putti. Mixed finite elements and Newton-type linearization for the solution of Richards' equation. *International journal for numerical methods in engineering*, 45:1025–1046, 1999.

[9] M. A. Celia, E. T. Bouloutas, and R. L. Zarba. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, 26(7):1483–1496, 1990.

[10] L. M. Chounet, D. Hilhorst, C. Jouron, Y. Kelanemer, and P. Nicolas. Simulation of water flow and heat transfer in soils by means of a mixed finite element method. *Advances in Water Resources*, 22(5):445–460, 1999.

[11] Todd S. Coffey. *Temporal and pseudo-temporal numerical integration methods*. PhD thesis, North Carolina State University, Raleigh, NC, 2002.

[12] I.S. Duff and H.A. van der Vorst. Developments and trends in the parallel solution of linear systems. *Parallel Computing*, 25(13–14):1931–1970, 1999.

[13] M. W. Farthing, C. E. Kees, and C. T. Miller. Mixed finite element methods and higher-order temporal approximations. *Advances in Water Resources*, 25(1):85–101, 2002.

[14] P.A. Forsyth, Y.-S. Wu, and K. Pruess. Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media. *Advances in Water Resources*, 18:25–38, 1995.

[15] W.D. Gropp, D.E. Keyes, L. C. McInnes, and M. D. Tidriri. Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit cfd. *International Journal of High Performance Computing Applications*, 14 (2):102–136, 2000.

[16] A Gupta. Recent advances in direct methods for solving unsymmetric sparse systems of linear equations. *Association for Computing Machinery, Transactions on Mathematical Software*, 28(3):301–324, 2002.

[17] E. W. Jenkins, C. E. Kees, C. T. Kelley, and C. T. Miller. An aggregation-based domain decomposition preconditioner for groundwater flow. *SIAM Journal on Scientific Computing*, 23(2):430–441, 2001.

[18] J.E. Jones and C.S. Woodward. Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. *Advances in Water Resources*, 24(7):763–774, 2001.

[19] D. Kavetski, P. Binning, and S.W. Sloan. Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards' equation. *Advances in Water Resources*, 24:595–605, 2001.

[20] C. E. Kees and C. T. Miller. C++ implementations of numerical methods for solving differential-algebraic equations: Design and optimization considerations. *Association for Computing Machinery, Transactions on Mathematical Software*, 25(4):377–403, 1999.

[21] C.E. Kees and C.T. Miller. Higher order time integration methods for two-phase flow. *Advances in Water Resources*, 25(2):159–177, 2002.

[22] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.

[23] C.T. Kelley and D.E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35(2):508–523, 1998.

[24] D.A. Knoll and P.R. McHugh. Enhanced nonlinear iterative techniques applied to nonequilibrium plasma flow. *SIAM Journal on Scientific Computing*, 19(1):291–301, 1998.

[25] F. Lehmann and Ph. Ackerer. Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media. *Transport in Porous Media*, 31(3):275–292, 1998.

[26] C. T. Miller, G. Christakos, P. T. Imhoff, J. F. McBride, J. A. Pedit, and J. A. Trangenstein. Multiphase flow and transport modeling in heterogeneous porous media: Challenges and approaches. *Advances in Water Resources*, 21(2):77–120, 1998.

[27] C. T. Miller, G. A. Williams, C. T. Kelley, and M. D. Tocci. Robust solution of Richards' equation for non-uniform porous media. *Water Resources Research*, 34(10):2599–2610, 1998.

[28] Y. Mualem. A new model for predicting the hydraulic conductivity of

unsaturated porous media. *Water Resources Research*, 12(3):513–522, 1976.

[29] C. Paniconi and M. Putti. A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems. *Water Resources Research*, 30(12):3357–3374, 1994.

[30] J.-Y. Parlange, W.L. Hogarth, D.A. Barry, M.B. Parlange, R. Haverkamp, P.J. Ross, T.S. Steenhuis, D.A. DiCarlo, and G. Katul. Analytical approximations to the solution of Richards' equation with applications to infiltration, ponding, and time compression approximation. *Advances in Water Resources*, 23:189–194, 1999.

[31] K. Rathfelder and L. M. Abriola. Mass conservative numerical solutions of the head-based Richards equation. *Water Resources Research*, 30(9): 2579–2586, 1994.

[32] J. Simunek, T. Vogel, and M. Th. van Genuchten. The SWMS_2D code for simulating water flow and solute transport in two-dimensional variably saturated media. Technical Report Research Report 132, U. S. Salinity Laboratory, Agricultural Research Service, U. S. Department of Agriculture, Riverside, CA, 1994.

[33] M. D. Tocci, C. T. Kelley, and C. T. Miller. Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines. *Advances in Water Resources*, 20(1):1–14, 1997.

[34] M. D. Tocci, C. T. Kelley, C. T. Miller, and C. E. Kees. Inexact Newton methods and the method of lines for solving Richards' equation in two space dimensions. *Computational Geosciences*, 2(4):291–309, 1999.

[35] M. Th. van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44:892–898, 1980.

[36] G. A. Williams, C. T. Miller, and C. T. Kelley. Transformation approaches for simulating flow in variably saturated porous media. *Water Resources Research*, 36(4):923–934, 2000.

[37] Y.-S. Wu and P.A. Forsyth. On the selection of primary variables in numerical formulation for modeling multiphase flow in porous media. *Journal of Contaminant Hydrology*, 48:277–304, 2001.

[38] J. Zaidel and D. Russo. Estimation of finite difference interblock conductivities for simulation of infiltration into initially dry soils. *Water Resources Research*, 28(9):2285–2295, 1992.