

Locally conservative, stabilized finite element methods for variably saturated flow

C.E. Kees* and M.W. Farthing

Coastal and Hydraulics Laboratory, US Army Engineer Research and Development Center, 3909 Halls Ferry Rd., Vicksburg, MS 39180-6133

C.N. Dawson

Center for Subsurface Modeling, Institute for Computational Engineering and Sciences, The University of Texas at Austin 1 University Station, C0200 Austin, TX 78712

Abstract

Standard Galerkin finite element methods for variably saturated groundwater flow have several deficiencies. For instance, local oscillations can appear around sharp infiltration fronts without the use of mass-lumping, and velocity fields obtained from differentiation of pressure fields are discontinuous at element boundaries. Here, we consider conforming finite element discretizations based on a multiscale formulation along with recently developed, local postprocessing schemes. The resulting approach maintains the basic flexibility and appeal of traditional finite element methods, while controlling nonphysical oscillations and producing element-wise mass-conservative velocity fields. Accuracy and efficiency of the proposed schemes are evaluated through a series of steady-state and transient variably saturated groundwater flow problems in homogeneous as well as heterogeneous domains.

Key words: Richards' equation, finite element method, multiscale stabilization, local conservation

1 Introduction

Richards' equation is a widely studied nonlinear parabolic equation describing water flow in variably saturated porous media [1]. Analytical solutions for

* Corresponding Author.

Email address: christopher.e.kees@erdc.usace.army.mil (C.E. Kees).

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE NOV 2007		2. REPORT TYPE		3. DATES COVERED 00-00-2007 to 00-00-2007	
4. TITLE AND SUBTITLE Locally conservative, stabilized finite element methods for variably saturated flow				5a. CONTRACT NUMBER	
				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) US Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory, 3909 Halls Ferry Rd, Vicksburg, MS, 39180-6133				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					
14. ABSTRACT Standard Galerkin finite element methods for variably saturated groundwater flow have several deficiencies. For instance, local oscillations can appear around sharp infiltration fronts without the use of mass-lumping, and velocity fields obtained from differentiation of pressure fields are discontinuous at element boundaries. Here, we consider conforming finite element discretizations based on a multiscale formulation along with recently developed, local postprocessing schemes. The resulting approach maintains the basic flexibility and appeal of traditional finite element methods, while controlling nonphysical oscillations and producing element-wise mass-conservative velocity fields. Accuracy and efficiency of the proposed schemes are evaluated through a series of steady-state and transient variably saturated ground-water flow problems in homogeneous as well as heterogeneous domains.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Richards' equation are limited in number and applicability due to its non-linearity and the degree of spatial heterogeneity found in many problems of interest [2]. Numerical solutions of Richards' equation are therefore important for modeling groundwater flow and contaminant transport in the subsurface.

The qualities that make it difficult, if not impossible, to solve analytically also contribute to the challenge of developing accurate and efficient numerical methods for Richards' equation. In order to be robust, spatial and temporal approximations must be capable of resolving sharp fronts infiltrating dry media and handling transitions to elliptic or nearly elliptic conditions in saturated regions [3]. Regardless of the discretization used, the result is almost always a large nonlinear system that is hard to solve efficiently [4, 5].

While finite difference, finite volume, and finite element approximations are all regularly used for Richards' equation [6–8], the focus here is on finite element methods (FEMs), which are appealing because they are well-suited for unstructured spatial meshes and so can be readily applied to irregular domains and facilitate some classes of adaptive refinement [9]. Classical FEMs do not perform well for Richards' equation. Most notably, lumping and upwinding schemes are required to prevent significant non-physical oscillations [6, 10] and straightforward evaluation of Darcy's law leads to an approximate groundwater velocity field that is not locally conservative over mesh elements. By this we mean that normal fluxes are discontinuous across element boundaries, and the velocity field's divergence over an element does not balance the discrete mass accumulation [11].

Many remedies to these deficiencies have been proposed. Mass lumping and upwinding are commonly introduced to smear steep transition zones, while postprocessing schemes or mixed FEM formulations have been considered to obtain locally conservative velocity fields [6, 10, 12–14]. An exhaustive review of available discretization techniques is beyond the scope of this work. A review of recent methods can be found in [15]

The purpose of this work is to evaluate two complementary strategies for improving conforming Galerkin (CG) FEM approximations of Richards' equation on simplicial meshes in one, two, or three dimensions. Specifically, we consider a multiscale, stabilized finite element formulation [16] and a pair of local postprocessing techniques from [17] and [18] that produce element-wise conservative velocity fields. The approaches are also orthogonal in the sense that the velocity postprocessing methods are applicable for generic CG approximations [17] or fairly general data sets [18], and the stabilized solution is not restricted to the proposed postprocessing [19]. On the other hand, the combined approach is appealing, since it maintains the basic flexibility and simplicity of traditional FEMs while controlling nonphysical oscillations and producing element-wise conservative velocity fields.

This paper is organized as follows. In the next section we review relevant work on FEMs and locally conservative velocity approximations. In §3 we summarize our formulation of Richards’ equation, while multiscale stabilized CG methods based on the work of [20] are described in §4.1 – §4.3, and algorithms for obtaining locally conservative approximations based on [17, 18] are presented in §4.4. Finally, in §5 we provide numerical results comparing these methods using several error and computational work measures, followed by discussion and conclusions.

2 Background

2.1 Finite element approximations for Richards’ equation

A variety of FEM-based schemes have been applied to Richards’ equation. For instance, [6] presented a CG method for Richards’ equation based on modified Picard linearization for temporal derivatives and mass-lumping, which corresponds to a reduced order quadrature formula for the mass accumulation term. The resulting approximation is globally conservative with good control of spurious numerical oscillations. Along with the similar “chord-slope” technique from [21], this can be considered the standard approach for treating accumulation terms in Richards’ equation. Later, [10] demonstrated that mass-lumping combined with appropriate “upwinded” relative permeability evaluation can lead to monotone approximations independent of mesh resolution.

Standard CG schemes do not directly yield element-wise conservative velocity fields as defined above, and an additional postprocessing step is needed [22, 23, 47]. This complication is avoided by finite element formulations that include an explicit velocity representation, such as mixed FEMs and local discontinuous Galerkin (LDG) methods [24, 25]. While they have proven successful for many subsurface flow problems including Richards’ equation [12, 14, 26], mixed FEMs are not without drawbacks. The resulting linear systems are saddle point problems without hybridization [24], and the number of unknowns is greater than a nodal CG approximation of the same order on the same mesh [25]. As with CG methods, overshoot and undershoot can also occur for sharp front problems without upwinding and reduced order quadrature approximations [4]. Similarly, LDG approximations can exhibit overshoot and undershoot when direct evaluation of relative hydraulic conductivities is used without upwinding [3]. It should be noted that this behavior is a function of grid resolution. For sufficiently refined meshes, oscillations are eliminated. Sufficient resolution is, however, unreachable in many cases without the use of adaptive mesh refinement [12, 27].

A thorough evaluation of the relative merits of mixed and CG FEM discretizations for Richards’ equation is beyond the scope of this work. Direct comparisons between a mixed hybrid FEM and a traditional, local velocity postprocessing technique [28] for saturated groundwater flow can be found in [29–31]. Broadly speaking, the added degrees of freedom typically associated with a mixed method must be weighed against the accuracy and generality of a given postprocessing technique for CG FEMs. Moreover, these approaches are not necessarily mutually exclusive (see for instance the combination of CG and LDG approximations in [25]). In our view, a fully satisfactory approach has not been reached, since existing mixed and CG discretizations require some type of lumping and reduced order approximation to avoid oscillation.

2.2 Variational multiscale finite element methods

Upwinding and mass lumping have been recognized for some time as an attempt to properly account for “subgrid” effects in discretizations [32]. From this perspective, it is unresolved features in the solution that produce qualitatively incorrect features such as spurious local maxima and minima. Stabilized methods like the streamline upwind Petrov Galerkin (SUPG) discretization were originally introduced as artificial viscosity methods that attempted to introduce minimal amounts of numerical viscosity to control oscillations on coarse meshes [33]. In the last twenty years, stabilized FEMs have been widely used within the computational fluid dynamics community for modeling advection-diffusion-reaction systems as well as incompressible Navier-Stokes problems [34–36].

Residual-based, stabilized finite elements were recast as the multiscale variational method in [20]. This provided a framework for relating unresolved solution components and stabilization terms used to supplement standard Galerkin formulations. Variational multiscale-based approximations have since become increasingly popular [16, 37, 38] and were applied to two-phase flow in porous media in [39] as well as [40].

The focus of traditional stabilized methods was improved numerical approximation, particularly the resolution of internal or boundary layers [33, 41]. This is also the perspective taken in [40]. The emphasis of [42] as well as the related multiscale finite element and finite volume methods from [43–46] is on incorporation of fine-scale heterogeneity into the resolved (or coarse-scale) solution. In the development below, we will focus on improved resolution of sharp fronts, since this is a particular challenge in extending numerical methods for saturated flow to Richards’ equation. To our knowledge the variational multiscale framework has not been applied to Richards’ equation, nor has the issue of obtaining locally conservative groundwater velocity fields from stabilized,

conforming FEMs been addressed.

2.3 Velocity postprocessing techniques

As mentioned above, standard CG methods for groundwater flow do not directly yield locally conservative velocity approximations. A global postprocessing technique can be found in [13] while a local postprocessing algorithm based on dual meshes and stream functions can be found in [28]. On the other hand, the recent popularity of discontinuous Galerkin methods has renewed interest in local conservation for CG discretizations. The general conservation properties of CG methods for advection-diffusion-reaction equations and incompressible Navier-Stokes were investigated in [19, 22]. Furthermore a general method for obtaining locally conservative velocities from standard CG solutions was outlined, which required the solution of a global linear system.

Here, we focus on two recent postprocessing schemes from [17] and [18] that can be computed locally and are quite general in their domain of applicability. The two methods are related, since they arise from equivalent global formulations based on piecewise constant corrections to a velocity field that is singly defined on element boundaries, but not locally conservative element-wise [17, 18]. The approaches differ in that [17] focuses explicitly on velocities obtained from CG solutions, while [18] addresses more general velocity fields (e.g., non-conservative velocity fields arising from field measurements as well as numerical methods).

The approach taken in [17] is based on the idea of enriching the solution space with additional discontinuous degrees of freedom after a CG solution has been obtained. The CG solution is modified strictly to satisfy the local conservation property, which results in a postprocessing method that depends only on the mesh and information left over from the CG solve. A local approach [17, algorithm 2] is derived by using piecewise linear enrichment and domain decomposition. The result is a locally conservative velocity field defined on element boundaries (i.e., a numerical trace) after directly solving small linear systems over the mesh node stars, which are simply the collection of elements sharing a given node for each node in the mesh.

The perspective taken in [18] is to minimize the correction necessary to obtain local conservation in a suitably chosen norm. A Gauss-Seidel like iteration is then derived which requires strictly local updates using coefficients that depend only on mesh information. The overhead for an iteration of the Sun-Wheeler Gauss-Seidel approach is less than that of [17], since it uses element interface connectivity and does not require solution of local systems. On the other hand, the algorithm is iterative with a convergence rate that depends

on the mesh [18].

In both approaches, the output is a locally conservative velocity field defined on element boundaries. This may be sufficient for many applications, like use of the velocity for a finite volume transport calculation. Alternatively, the element boundary data can be used to define a local Neumann problem on each element that can then be solved with a mixed finite element space to obtain a conservative, globally defined velocity field. In some cases, like the lowest order Raviart-Thomas space, the local step reduces to a simple projection [18]. With a global mixed FEM velocity representation, not only local conservation but also higher order compatibility with transport discretizations, in the sense of [47], is then available.

2.4 Nonconforming finite element approximation

To better evaluate accuracy and computational expense for the postprocessing techniques from §4.4.1 and §4.4.2, we would like to compare them to velocities obtained from mixed finite element approximations, since these are well-established, locally conservative methods [24, 48]. For simplicity, we follow an approach similar to that taken in [49] and exploit a correspondence between P^1 nonconforming and mixed hybrid FEM approximations [50].

To be more specific, a very simple postprocessing step was shown in [51] to give correspondence between the standard P^1 nonconforming approximation and a mixed hybrid FEM solution for elliptic problems with isotropic conductivity and a piecewise constant data assumption. Arbogast et. al [52] showed correspondence between mixed and nonconforming methods for a larger class of elliptic problems including lower order terms. In general, the correspondence requires supplementing standard nonconforming spaces with bubbles and a series of L_2 projections in the weak formulation. On the other hand, [53] presented a postprocessing approach in which the RT0 velocity and potential could be obtained via postprocessing of P^1 nonconforming solutions for the potential variable without reference to bubble functions in the case of full tensor conductivities as well as quasi-linear elliptic equations. This is the approach followed below for comparison with the CG velocity postprocessing techniques.

3 Richards' equation

We begin with a mass conservation statement for the aqueous phase in an air-water system where the solid phase is assumed to be immobile and interphase

mass transfer is negligible

$$\begin{aligned} m_t + \nabla \cdot (\rho \mathbf{q}) &= b, \text{ for } \mathbf{x}, t \in \Omega \times [0, T] \\ m &= \rho \theta \end{aligned} \quad (1)$$

along with the standard extension of Darcy's law to variably saturated conditions [54]

$$\mathbf{q} = -k_r \mathbf{K}_s (\nabla \psi - \rho \mathbf{g}_u) \quad (2)$$

Here ψ and θ are the pressure head and volume fraction, respectively, and $\rho = \varrho/\varrho_0$ is a normalized density. The relative permeability, k_r is assumed to be a scalar, but the saturated conductivity \mathbf{K}_s is not. b is a generic source term, and \mathbf{g}_u is a unit vector accounting for the direction of gravity [11]. Finally, Ω represents the spatial domain and $[0, T]$ the time interval of interest.

An equation of state for ϱ and constitutive (p - s - k) relations are necessary to close eqns (1) and (2). In the following, we assume that the aqueous phase density can be written

$$\varrho = \varrho_0 e^{\beta_c(\psi - \psi_0)} \quad (3)$$

for a reference pressure head, ψ_0 . We also adopt standard van Genuchten [55] and Mualem [56] relations to describe the interdependence of fluid pressure, saturation, and the relative permeability, although the approaches considered are not restricted by these choices. For $\psi < 0$, the van Genuchten-Mualem p - s - k 's are

$$\begin{aligned} s_e &= \frac{(\theta - \theta_r)}{(\theta_s - \theta_r)} \\ s_e &= [1 + (\alpha_{vg} |\psi|)^{n_{vg}}]^{-m_{vg}} \\ k_r &= \sqrt{s_e} \left[1 - (1 - s_e^{1/m_{vg}})^{m_{vg}} \right]^2 \end{aligned} \quad (4)$$

where s_e is the effective saturation, θ_r is the residual volumetric water content, θ_s is the saturated volumetric water content, α_{vg} is a parameter related to the mean pore size, n_{vg} is a parameter related to the uniformity of the soil pore-size distribution, and $m_{vg} = 1 - 1/n_{vg}$. For $\psi \geq 0$, the porous medium is fully saturated, and eqn (4) reverts to $s_e = 1$ and $k_r = 1$.

Initial and boundary conditions for eqns (1) and (2) are written

$$\psi(\mathbf{x}, 0) = \psi^0 \quad (5)$$

$$\psi(\mathbf{x}, t) = \psi^b, \mathbf{x} \in \Gamma_D \quad (6)$$

$$\rho \mathbf{q} \cdot \mathbf{n} = \sigma^b, \mathbf{x} \in \Gamma_N \quad (7)$$

where $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$, and \mathbf{n} is the unit outer normal on $\partial\Omega$.

Several variations of eqns (1) and (2) are common depending on the choice of solution variable (ψ or θ) and the manner in which the accumulation term in eqn (1) is handled. Here, we follow the basic approach outlined in [7], which can be seen as a generalization of the standard mixed form [6]. That is, we take ψ as the dependent variable, apply temporal approximations directly to m_t rather than using the chain rule, and rely upon the nonlinear solution approach to resolve the dependence of m on ψ [7, 11].

4 Numerical methods

Starting from the basic mass-conservative formulation in §3, we describe a multiscale-stabilized finite element method for Richards' equation below. We then present straightforward extensions of postprocessing algorithms from [17] and [18] to obtain locally conservative velocity fields from CG solutions of nonlinear scalar parabolic PDEs. Lastly, we summarize our approaches for time integration and solving the linear and nonlinear systems that arise from the multiscale finite element discretization.

4.1 Weak formulation

In an attempt to simplify presentation of the numerical methods below, we first substitute eqn (2) into eqn (1) and write the result as a generic nonlinear advection-diffusion equation

$$m_t + \nabla \cdot (\mathbf{f} - \mathbf{a} \nabla \psi) = b \quad (8)$$

where

$$\begin{aligned} \mathbf{f} &= \rho^2 k_r \mathbf{K}_s \mathbf{g}_u, \\ \mathbf{a} &= \rho k_r \mathbf{K}_s \\ \boldsymbol{\sigma} &= \mathbf{f} - \mathbf{a} \nabla \psi = \rho \mathbf{q} \end{aligned} \quad (9)$$

We then apply Hughes' variational multiscale paradigm [20] to eqn (8) in a manner similar to the approach taken in [40]. To begin, trial solutions are

sought in

$$V = \{v \in H^1(\Omega) : v = \psi^b \text{ on } \Gamma_D\} \quad (10)$$

while test functions are taken from

$$W = \{w \in H^1(\Omega) : w = 0 \text{ on } \Gamma_D\} \quad (11)$$

where $H^1(\Omega)$ is the Sobolev space of functions that are square integrable and have square-integrable first derivatives over Ω . For $t \in [0, T]$ fixed, a solution $\psi \in V$ is sought such that

$$\int_{\Omega} [m_t + \nabla \cdot (\mathbf{f} - \mathbf{a}\nabla\psi) - b] w \, dx = 0, \quad (12)$$

$$\int_{\Omega} \psi(\mathbf{x}, 0) w \, dx = \int_{\Omega} \psi^0(\mathbf{x}) w \, dx, \quad \forall w \in W \quad (13)$$

Integrating eqn (14) by parts gives the weak statement, find $\psi \in V$ such that

$$\begin{aligned} \int_{\Omega} m_t w \, dx - \int_{\Omega} (\mathbf{f} - \mathbf{a}\nabla\psi) \cdot \nabla w \, dx + \int_{\Gamma_N} \sigma^b w \, ds - \int_{\Omega} b w \, dx \\ = 0, \quad \forall w \in W \end{aligned} \quad (14)$$

4.2 Temporal approximation

Before applying a finite element approximation we discretize in time, following [57]. Specifically, we use fully implicit backwards difference formula (BDF) methods [58] in which all terms in eqn (14) are approximated at the new time level t^{n+1} and the accumulation term is approximated as

$$m_t \approx \hat{m}_t = \alpha^{n+1} m(\psi^{n+1}) + \beta^n \quad (15)$$

Here α^{n+1} depends on the order of approximation and time step history, while β depends on the order of approximation, time step history, and solution history. For instance, with a first order (backward Euler) approximation, $\alpha = 1/\Delta t^{n+1}$, $\beta^n = -m(\psi^n)/\Delta t^{n+1}$ for $\Delta t^{n+1} = t^{n+1} - t^n$.

4.3 Multiscale finite element approximation

In order to present the multiscale formulation, we insert some notation. Let T_h be a simplicial triangulation of Ω in \mathbb{R}^{n_d} , $n_d = 2, 3$, containing N_e elements, $\{\Omega_e\}$, $e = 1, \dots, N_e$, N_f element boundaries, or faces, $\{\gamma_f\}$, $f = 1, \dots, N_f$, and N_n nodes, $\{\mathbf{x}_n\}$, $n = 1, \dots, N_n$. The collection of faces in the domain interior is denoted Γ_I . We also assume that the intersection of elements $\Omega_e, \Omega_{e'} \in T_h$ is either empty, a unique $\gamma_f \in \Gamma_I$, an edge (for \mathbb{R}^3), or a point. The diameter of Ω_e is h_e and its unit outer normal is written \mathbf{n}_e .

The multiscale view of stabilization involves splitting V and W into resolved and unresolved scales

$$V = V_h \oplus \delta V \tag{16}$$

$$W = W_h \oplus \delta W \tag{17}$$

In our case V_h and W_h are just the usual conforming piecewise linear Galerkin spaces

$$V_h = \{v_h \in V \cap C^0(\bar{\Omega}) : v_h|_{\Omega_e} \in P^1(\Omega_e)\} \tag{18}$$

$$W_h = \{w_h \in W \cap C^0(\bar{\Omega}) : w_h|_{\Omega_e} \in P^1(\Omega_e)\} \tag{19}$$

while δV and δW remain infinite dimensional. The solution is then written uniquely as $\psi_h + \delta\psi$, and the subgrid error, $\delta\psi$, is approximated using local problems over each element. Note that we also assume that the boundary and initial data, eqns (5)–(7), can be accurately approximated at the grid-scale.

Inserting eqns (15) and (19) into eqn (14) and taking advantage of its linearity in w allows us to write a coupled problem for the solution $\psi_h + \delta\psi$ at t^{n+1}

$$\begin{aligned} F_h &= \int_{\Omega} \hat{m}_t w_h \, dx - \int_{\Omega} (\mathbf{f} - \mathbf{a} \nabla \psi) \cdot \nabla w_h \, dx + \int_{\Gamma_N} \sigma^b w_h \, ds - \int_{\Omega} b w_h \, dx \\ &= 0, \quad \forall w_h \in W_h \end{aligned} \tag{20}$$

$$\begin{aligned} F_{\delta} &= \int_{\Omega} \hat{m}_t \delta w \, dx - \int_{\Omega} (\mathbf{f} - \mathbf{a} \nabla \psi) \cdot \nabla \delta w \, dx + \int_{\Gamma_N} \sigma^b \delta w \, ds - \int_{\Omega} b \delta w \, dx \\ &= 0, \quad \forall \delta w \in \delta W \end{aligned} \tag{21}$$

where we have dropped time level indicators for convenience, with the understanding that all terms are evaluated at t^{n+1} . Before proceeding, we point out that our goal is to obtain a modified version of eqn (20)

$$G_h = F_h - \sum_e \int_{\Omega_e} \mathcal{L}_{s,h}^* w_h \tau \mathcal{R}_h(\psi_h) dx = 0, \forall w_h \in W_h \quad (22)$$

and a corresponding linearized system of equations to use in a Newton solution algorithm for ψ_h . Here, \mathcal{R}_h is an approximation to the strong residual

$$\mathcal{R} = \hat{m}_t + \nabla \cdot (\mathbf{f} - \mathbf{a} \nabla \psi) - b \quad (23)$$

and $\mathcal{L}_{s,h}^*$ approximates the formal adjoint of a linear operator \mathcal{L}_s defined below.

In order to linearize eqns (20) and (21), we label the Newton increment $v = v_h + \delta v$ and seek an approximate solution $\psi^+ = \psi_h^- + v$. We linearize around the coarse-scale quantity ψ_h^- at each iteration and assume $\delta v^- = 0$ [40]. The coupled, linearized problem is

$$\int_{\Omega} \hat{m}'_t v w_h dx - \int_{\Omega} (\mathbf{f}' v - \mathbf{a}' \nabla \psi_h^- v - \mathbf{a} \nabla v) \cdot \nabla w_h dx = -F_h^- \quad (24)$$

$\forall w_h \in W_h$

$$\int_{\Omega} \hat{m}'_t v \delta w dx - \int_{\Omega} (\mathbf{f}' v - \mathbf{a}' \nabla \psi_h^- v - \mathbf{a} \nabla v) \cdot \nabla \delta w dx = -F_{\delta}^- \quad (25)$$

$\forall \delta w \in \delta W$

Here, the ' symbol represents differentiation with respect to ψ and nonlinearities are evaluated at ψ_h^- . Below, a superscript $-$ is used to denote evaluation at ψ_h^- where necessary to avoid confusion.

4.3.1 Subgrid-scale approximation

We make a number of modeling assumptions to obtain a computationally tractable subgrid-scale approximation. First, we take a domain decomposition approach to obtain local (element) problems for δv . This is accomplished by assuming $\delta w = \delta v = 0$ on $\partial\Omega_e$ for all e . On each Ω_e , eqn (25) can then be written

$$\begin{aligned} & \int_{\Omega_e} \hat{m}'_t \delta v \delta w dx - \int_{\Omega_e} (\mathbf{f}' \delta v - \mathbf{a}' \nabla \psi_h^- \delta v - \mathbf{a} \nabla \delta v) \cdot \nabla \delta w dx = \\ & - \int_{\Omega_e} \hat{m}'_t v_h \delta w dx + \int_{\Omega_e} (\mathbf{f}' v_h - \mathbf{a}' \nabla \psi_h^- v_h - \mathbf{a} \nabla v_h) \cdot \nabla \delta w dx - F_{\delta,e}^- \end{aligned} \quad (26)$$

where

$$F_{\delta,e} = \int_{\Omega_e} \hat{m}_t \delta w \, dx - \int_{\Omega_e} (\mathbf{f} - \mathbf{a} \nabla \psi_h) \cdot \nabla \delta w \, dx - \int_{\Omega_e} b \delta w \, dx \quad (27)$$

Integrating the second term on the right hand side by parts gives

$$\begin{aligned} & \int_{\Omega_e} \hat{m}'_t \delta v \delta w \, dx - \int_{\Omega_e} (\mathbf{f}' \delta v - \mathbf{a}' \nabla \psi_h^- \delta v - \mathbf{a} \nabla \delta v) \cdot \nabla \delta w \, dx = \\ & - \int_{\Omega_e} \hat{m}'_t v_h \delta w \, dx - \int_{\Omega_e} \nabla \cdot [\mathbf{f}' v_h - \mathbf{a}' \nabla \psi_h^- v_h - \mathbf{a} \nabla v_h] \delta w \, dx - F_{\delta,e}^- \end{aligned} \quad (28)$$

which motivates definition of the linearized operator

$$\begin{aligned} \mathcal{L}v &= \hat{m}'_t v + \nabla \cdot [\mathbf{f}' v - \mathbf{a}' \nabla \psi_h^- v - \mathbf{a} \nabla v] \\ &= \hat{m}'_t v + \mathcal{L}_s v \end{aligned} \quad (29)$$

After again integrating $F_{\delta,e}$ and the left hand side by parts, eqn (28) can be written

$$\int_{\Omega_e} \mathcal{L} \delta v \delta w \, dx = - \int_{\Omega_e} \mathcal{L} v_h \delta w \, dx - \int_{\Omega_e} \mathcal{R}(\psi_h^-) \delta w \, dx \quad (30)$$

where $\mathcal{L}v_h$ can be understood as a grid-scale linearization of $\mathcal{R}(\psi_h^-)$ on each Ω_e . To simplify evaluation of \mathcal{L} and \mathcal{R} , we consider using the nonconservative approximations

$$\mathcal{R}_h = \hat{m}_t + \mathbf{f}' \cdot \nabla \psi_h - \mathbf{a}' \nabla \psi_h \cdot \nabla \psi_h - b \quad (31)$$

$$\mathcal{L}_h v_h = \hat{m}'_t v_h + \mathbf{f}' \cdot \nabla v_h - \mathbf{a}' \nabla \psi_h \cdot \nabla v_h \quad (32)$$

where coefficients are evaluated pointwise using ψ_h^- on each Ω_e and second order terms are dropped, since P^1 trial functions are used. Conservative approximations could also be formulated, but as the approximations above are simple and do not affect the conservation properties of the grid-scale discretization, we use them for this work.

4.3.2 Grid-scale equation

Returning to eqn (24), we collect terms for v_h and δv and write

$$\int_{\Omega} \hat{m}'_t v_h w_h \, dx - \int_{\Omega} (\mathbf{f}' v_h - \mathbf{a}' \nabla \psi_h^- v_h - \mathbf{a} \nabla v_h) \cdot \nabla w_h \, dx$$

$$\begin{aligned}
& + \sum_e \int_{\Omega_e} \hat{m}'_t \delta v w_h \, dx - \sum_e \int_{\Omega_e} \left(\mathbf{f}' \delta v - \mathbf{a}' \nabla \psi_h^- \delta v - \mathbf{a} \nabla \delta v \right) \cdot \nabla w_h \, dx \\
& \qquad \qquad \qquad = -F_h^-
\end{aligned}$$

Although not strictly necessary, we neglect the effects of subgrid scale variation on the grid-scale accumulation term (a “static subgrid scales” assumption [38]) and integrate the fourth term on the left hand side by parts to obtain

$$\begin{aligned}
& \int_{\Omega} \hat{m}'_t v_h w_h \, dx - \int_{\Omega} \left(\mathbf{f}' v_h - \mathbf{a}' \nabla \psi_h^- v_h - \mathbf{a} \nabla v_h \right) \cdot \nabla w_h \, dx \\
& \qquad \qquad \qquad + \sum_e \int_{\Omega_e} \mathcal{L}_s^* w_h \delta v \, dx = -F_h^-
\end{aligned} \tag{33}$$

where boundary terms

$$\sum_e \int_{\partial \Omega_e} \mathbf{a} \nabla w_h \cdot \mathbf{n}_e \delta v \, ds \tag{34}$$

in the integration by parts have again been neglected in the introduction of the formal adjoint of \mathcal{L}_s

$$\mathcal{L}_s^* w = - \left(\mathbf{f}' - \mathbf{a}' \nabla \psi_h^- \right) \cdot \nabla w - \nabla \cdot \left(\mathbf{a} \nabla w \right) \tag{35}$$

As with \mathcal{L}_h , we drop second order terms in eqn (35) and approximate \mathcal{L}_s^* as

$$\mathcal{L}_{s,h}^* w_h = -\mathbf{f}' \cdot \nabla w - \mathbf{a}' \nabla \psi_h^- \cdot \nabla w \tag{36}$$

To obtain a system in terms of the grid-scale quantities alone, we use a simple algebraic subgrid scale (ASGS) approximation [16, 33] and again neglect temporal variation in the subgrid scale Newton correction [38]. We then insert

$$\delta v \approx -\tau \left(\mathcal{L}_h v_h + \mathcal{R}_h^- \right) \tag{37}$$

into eqn (33) to obtain a linearized, grid-scale equation

$$\begin{aligned}
& \int_{\Omega} \hat{m}'_t v_h w_h \, dx - \int_{\Omega} \left(\mathbf{f}' v_h - \mathbf{a}' \nabla \psi_h^- v_h - \mathbf{a} \nabla v_h \right) \cdot \nabla w_h \, dx \\
& - \sum_e \int_{\Omega_e} \mathcal{L}_{s,h}^* w_h \tau \mathcal{L}_h v_h \, dx = -F_h^- + \sum_e \int_{\Omega_e} \mathcal{L}_{s,h}^* w_h \tau \mathcal{R}_{h,e}^- \, dx
\end{aligned} \tag{38}$$

The right hand side of eqn (38) is just $-G_h$ from eqn (22). The left hand side contains the standard conforming Galerkin terms for the Newton Jacobian plus an additional term arising from the stabilization. This yields a quasi-Newton method, since the linear operator on the left hand side is not precisely the Jacobian of G_h due to the linearization of the stabilization terms.

4.3.3 Stabilization parameter

There are a number of ways to define the stabilization parameter τ . Typically, these are motivated by analysis of linear advection-diffusion-reaction equations [33, 59]. For the 1D linear advection-diffusion equation with constant coefficients, the solution of the subgrid error approximation yields

$$\tau = \frac{h_e}{2a'} \left[\coth \left(\frac{h_e f'}{2a'} \right) - \frac{2a'}{h_e f'} \right] \quad (39)$$

which was already widely used in earlier stabilized methods [20]. Here, we use a straightforward multi-dimensional approximation of τ on Ω_e , similar to that used in [33, 40, 59],

$$\tau = \left[\left(2 \frac{\|\mathbf{f}' - \mathbf{a}' \nabla \psi_h\|_2}{h_e} \right)^2 + 9 \left(4 \frac{\|\mathbf{a}\|_\infty}{h_e^2} \right)^2 \right]^{-\frac{1}{2}} \quad (40)$$

where $\|\cdot\|_2$ is the vector 2-norm and $\|\cdot\|_\infty$ is the matrix ∞ -norm.

4.3.4 Shock capturing diffusion

The ASGS approximation above does not in general ensure that no spurious local extrema will be generated. Particularly for nonlinear problems, steep gradients in the solution can form which produce small undershoot or overshoot near the front. In such cases, it is common to include a shock capturing numerical diffusion term so that the grid-scale residual equation becomes

$$\begin{aligned} G_h &= F_h - \sum_e \int_{\Omega_e} \mathcal{L}_{s,h}^* w_h \tau \mathcal{R}_h(\psi_h) \, dx + \sum_e \int_{\Omega_e} \nu \nabla \psi_h \cdot \nabla w_h \, dx \\ &= 0, \forall w_h \in W_h \end{aligned} \quad (41)$$

There are several different approaches for defining the associated numerical diffusion parameter [60], but here we simply use the standard isotropic definition [60, 61] for each Ω_e

$$\nu = \nu_c \frac{h_e}{2} \frac{|\mathcal{R}_{h,e}|}{\|\nabla\psi_h\|_2} \quad (42)$$

where ν_c is a problem dependent parameter.

4.4 Velocity postprocessing algorithms

The goal of our postprocessing is to obtain a velocity field $\hat{\boldsymbol{\sigma}}_h$ that conserves mass discretely on each Ω_e , $e = 1, \dots, N_e$, has continuous normal component across each $\gamma_f \in \Gamma_I$, and satisfies the boundary condition eqn (7) along Γ_N . To do this, we extend the techniques of Larson and Niklasson [17] and Sun and Wheeler [18] to eqns (8) and (9) in a straightforward manner.

4.4.1 Larson-Niklasson [17] postprocessing

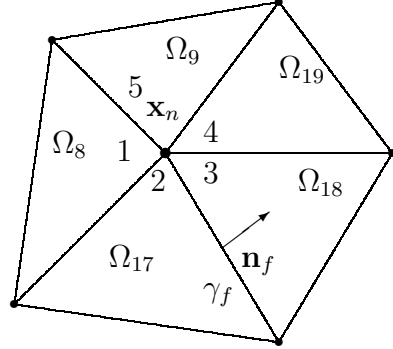
Before proceeding, some additional notation is required. The set of elements sharing node \mathbf{x}_n , or its star, is written $\mathcal{E}(n)$. The number of elements sharing \mathbf{x}_n is $N_{n,e}^* = \text{card}(\mathcal{E}(n))$. For each interior face $\gamma_f = \partial\Omega_\ell \cap \partial\Omega_r$, we write \mathbf{n}_ℓ and \mathbf{n}_r for the unit outer normal to Ω_ℓ and Ω_r , respectively. The global element identifiers for Ω_ℓ and Ω_r are also written $e_\ell(f)$ and $e_r(f)$. An arbitrary choice of $\mathbf{n}_f = \mathbf{n}_\ell$ is made in order to associate a unique unit normal vector with each γ_f . The set of faces contained in $\partial\Omega_e$ that are in $\Gamma_I \cup \Gamma_D$ is written $\mathcal{F}_{i,d}(e)$. The global identifiers for the nodes contained in a face γ_f are denoted $\mathcal{N}(f)$. A local numbering of elements $e \mapsto e^*$ for $\Omega_e \in \mathcal{E}(n)$ is also associated with each node-star. Figures 1 and 2 illustrate our notational conventions for a two-dimensional triangulation.

Since we consider only piecewise-linear approximations, global basis functions are naturally associated with nodes \mathbf{x}_n , $n = 1, \dots, N_n$. For each node, we then write the corresponding test function with $w_h(\mathbf{x}_n) = 1$ as $w_{h,n}$ and the discrete residual associated with each $e \in \mathcal{E}(n)$ as

$$\begin{aligned} G_{h,n,e} = & \int_{\Omega_e} \hat{m}_t w_{h,n} \, dx - \int_{\Omega_e} (\mathbf{f} - \mathbf{a} \nabla \psi_h) \cdot \nabla w_{h,n} \, dx + \int_{\partial\Omega_e \cap \Gamma_N} \sigma^b w_{h,n} \, ds \\ & - \int_{\Omega_e} b w_{h,n} \, dx - \int_{\Omega_e} \mathcal{L}_{s,h}^* w_{h,n} \tau \mathcal{R}_h(\psi_h) \, dx \\ & + \int_{\Omega_e} \nu \nabla \psi_h \cdot \nabla w_{h,n} \, dx \end{aligned} \quad (43)$$

In terms of the local numbering on $\mathcal{E}(n)$, the conservation residual associated with Ω_e is similarly

$$\begin{aligned}\mathcal{E}(n) &= \{8, 9, 17, 18, 19\} \\ e^* &= \{1, 5, 2, 3, 4\}\end{aligned}$$



$$\begin{aligned}e_\ell(f) &= 17, e_r(f) = 18 \\ e_\ell^*(f) &= 2, e_r^*(f) = 3\end{aligned}$$

Fig. 1. Notation example for interior node, \mathbf{x}_n

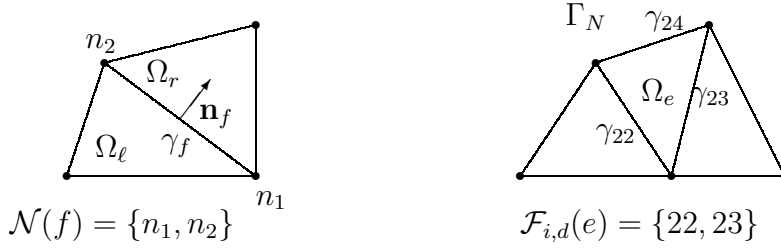


Fig. 2. Notation example for $\gamma_f \in \Gamma_I$ (left) and $\gamma_f \in \Gamma_N$ (right)

$$\begin{aligned}R_{n,e^*} &= G_{h,n,e} \\ &+ \sum_{f \in \mathcal{F}_{i,d}(e)} \int_{\gamma_f} [\bar{\boldsymbol{\sigma}}_{h,f} + (U_{n,e_\ell^*(f)} - U_{n,e_r^*(f)}) \mathbf{n}_f] w_{h,n} \cdot \mathbf{n}_e \, ds\end{aligned}\quad (44)$$

where $\bar{\boldsymbol{\sigma}}_{h,f}$ is the average velocity across γ_f given by

$$\bar{\boldsymbol{\sigma}}_{h,f} = \frac{(\mathbf{f} - \mathbf{a}\nabla\psi)|_{\Omega_{e_\ell(f)}} + (\mathbf{f} - \mathbf{a}\nabla\psi)|_{\Omega_{e_r(f)}}}{2}\quad (45)$$

and where $e_\ell^*(f)$ and $e_r^*(f)$ refer to the local numbering of the elements neighboring γ_f , and U_{n,e^*} , $e^* = 1, \dots, N_{n,e}^*$ are the piecewise constant corrections for which the algorithm solves.

At each node, we then have a system of $N_{n,e}^*$ equations from (44)

$$\mathbf{R}_n(\mathbf{U}_n) = \mathbf{0}\quad (46)$$

where $\mathbf{R}_n = [R_{n,1}, \dots, R_{n,N_{n,e}^*}]^T$ and $\mathbf{U}_n = [U_{n,1}, \dots, U_{n,N_{n,e}^*}]^T$. Solution of eqn (46) corresponds to a pure Neumann problem on $\mathcal{E}(n)$ when $\mathbf{x}_n \in \Gamma_I$, which is unique up to a constant [17]. To handle this non-uniqueness, we simply enforce $U_{n,1} = 0$ for nodes in Γ_I . Note that this arbitrary condition does not affect the resulting velocity, which is unique [17].

To solve eqn (46), we first compute $\mathbf{R}_n(\mathbf{0})$ which is just the conservation residual for $\bar{\boldsymbol{\sigma}}_h$. \mathbf{U}_n follows from

$$\mathbf{U}_n = -\mathbf{J}_n^{-1} \mathbf{R}_n(\mathbf{0}) \quad (47)$$

Here

$$J_{n,ij} = \frac{\partial R_{n,i}}{\partial U_{n,j}}, \quad i = 2, \dots, N_{n,e}^* \text{ and } j = 1, \dots, N_{n,e}^* \quad (48)$$

with $J_{n,1,j} = \delta_{1,j}$ and $R_{n,1} = 0$ for $\mathbf{x}_n \in \Gamma_I \cup \Gamma_N$ in order to enforce $U_{n,1} = 0$.

Eqn (47) is just a single step of Newton's method for eqn (46), which suffices since eqn (46) is linear in U_n even though it is nonlinear in ψ_h . Looking at eqn (44), the entries of \mathbf{J}_n are not dependent on the solution, coefficients, or boundary conditions, and depend only on the computational mesh for interior nodes. For nodes on $\partial\Omega$, \mathbf{J}_n depends on boundary condition types but not actual values. The local systems \mathbf{J}_n , $n = 1, \dots, N_n$ can be built and stored in factored form and only need updating when the mesh adapts or Γ_N changes.

Once a solution for U_{n,e^*} , $e^* = 1, \dots, N_n^*$ for each $n = 1, \dots, N_n$, has been obtained, the corrected velocity on each $\gamma_f \in \Gamma_I \cup \Gamma_D$ is

$$\hat{\boldsymbol{\sigma}}_{h,f} = \bar{\boldsymbol{\sigma}}_{h,f} + \sum_{n \in \mathcal{N}(f)} \left(U_{n,e_\ell^*(f)} - U_{n,e_r^*(f)} \right) \mathbf{n}_f w_{h,n} \quad (49)$$

with $U_{n,e_r^*(f)}$ omitted when $\gamma_f \in \Gamma_D$. We set $\hat{\boldsymbol{\sigma}}_{h,f} = \sigma^b \mathbf{n}_f$ for $\gamma_f \in \Gamma_N$.

$\hat{\boldsymbol{\sigma}}_{h,f} \cdot \mathbf{n}_f$ is continuous across γ_f by construction. To see that it is locally conservative, note that $R_{n,e} = 0$ for all n and e , and consider Ω_e with $\partial\Omega_e \cap \Gamma_N = \emptyset$

$$\begin{aligned} 0 &= \sum_{n \in \mathcal{N}(e)} R_{n,e} \\ &= \sum_{n \in \mathcal{N}(e)} G_{h,n,e} \\ &\quad + \sum_{n \in \mathcal{N}(e)} \sum_{f \in \mathcal{F}_{i,d}(e)} \int_{\gamma_f} \left[\bar{\boldsymbol{\sigma}}_{h,f} + \left(U_{n,e_\ell^*(f)} - U_{n,e_r^*(f)} \right) \mathbf{n}_f \right] w_{h,n} \cdot \mathbf{n}_e \, ds \end{aligned} \quad (50)$$

(51)

Since the test functions are a partition of unity, it follows that

$$\begin{aligned}
0 &= \int_{\Omega_e} (\hat{m}_t - b) \, dx \\
&\quad + \sum_{f \in \mathcal{F}_{i,d}(e)} \int_{\gamma_f} \left[\bar{\boldsymbol{\sigma}}_{h,f} + \sum_{n \in \mathcal{N}(e)} (U_{n,e_\ell^*}(f) - U_{n,e_r^*}(f)) \mathbf{n}_f w_{h,n} \right] \cdot \mathbf{n}_e \, ds \\
&= \int_{\Omega_e} (\hat{m}_t - b) \, dx + \sum_{f \in \mathcal{F}_{i,d}(e)} \int_{\gamma_f} \hat{\boldsymbol{\sigma}}_{h,f} \cdot \mathbf{n}_e \, ds
\end{aligned} \tag{52}$$

which gives the desired mass conservation statement. For elements on the Neumann boundary conservation follows likewise since $\hat{\boldsymbol{\sigma}}_{h,f} \cdot \mathbf{n}_e = \sigma^b$.

After obtaining $\hat{\boldsymbol{\sigma}}_{h,f}$ on each γ_f , $f = 1, \dots, N_f$ we obtain a representation over all of Ω by projecting $\hat{\boldsymbol{\sigma}}_{h,f}$ onto a local RT0 velocity space [52]

$$\hat{\mathbf{V}}_h(\Omega_e) = [P^0(\Omega_e)]^{n_d} \oplus \mathbf{x}P^0(\Omega_e) \tag{53}$$

Since $\hat{\boldsymbol{\sigma}}_{h,f}$ is piecewise linear with continuous normal component, the projection onto $\hat{\mathbf{V}}_h(\Omega_e)$ is well defined and the resulting global velocity field is in $H(\text{div}, \Omega)$. An RT0 representation is particularly convenient, since the normal flux through element faces

$$\int_{\gamma_f} \hat{\boldsymbol{\sigma}}_{h,f} \cdot \mathbf{n}_f \, ds \tag{54}$$

can be used as the degree of freedom along with the local basis [9]

$$\mathbf{N}_{e,i_f} = \frac{1}{n_d |\Omega_e|} (\mathbf{x} - \mathbf{x}_{n,i_f}), \quad i_f = 1, \dots, n_d + 1 \tag{55}$$

where i_f is a local identifier on Ω_e for γ_f and \mathbf{x}_{n,i_f} is the node across from face i_f .

In fact, one could also project $\hat{\boldsymbol{\sigma}}_{h,f}$ onto the linear Brezzi-Douglas-Duran-Fortin (BDDF1) space [62] (the Brezzi-Douglas-Marini space for $n_d = 2$), since $\hat{\boldsymbol{\sigma}}_{h,f}$ is piecewise linear. However, this requires more storage ($n_d^2 + n_d$ versus $n_d + 1$ local degrees of freedom), while only first order accuracy can be expected from $\hat{\boldsymbol{\sigma}}_h$ in general [17].

4.4.2 Sun-Wheeler [18] postprocessing

Although its original context is more general, we restate the Gauss-Seidel algorithm from [18] for the applications of interest here, CG solutions for Richards' equation. The notational conventions differ from the original presentation as well in order to be more consistent with the formulations above.

The Sun-Wheeler Gauss-Seidel algorithm assumes that an initial, well-defined velocity field $\bar{\sigma}_{h,f}$ is available and that $\bar{\sigma}_{h,f}$ is globally conservative. It then relies on constant element boundary corrections, $\{\Delta U_f\}$, $f = 1, \dots, N_f$. For given $\bar{\sigma}_{h,f}$ and $\{\Delta U_f\}$, we define the element conservation residual

$$R_e = \sum_{n \in \mathcal{N}(e)} G_{h,n,e} + \sum_{\gamma_f \in \partial\Omega_e} \int_{\gamma_f} (\bar{\sigma}_{h,f} + \Delta U_f \mathbf{n}_f) \cdot \mathbf{n}_e \, ds, \quad e = 1, \dots, N_e \quad (56)$$

and its average $\bar{R}_e = R_e/|\Omega_e|$. The assumption of global conservation implies that no correction is necessary for $\gamma_f \in \partial\Omega$, so we introduce the additional notation N_{f_i} for the number of interior element boundaries and a reordering for convenience so that $\gamma_f \in \Gamma_I$ for $1 \leq f \leq N_{f_i}$, $\gamma_f \in \partial\Omega$ for $N_{f_i} < f \leq N_f$, and $\Delta U_f = 0$ for $f > N_{f_i}$.

The algorithm is build around the idea of minimizing the L_2 norm of \bar{R}

$$\left[\int_{\Omega} \bar{R}^2 \, dx \right]^{1/2} \quad (57)$$

with respect to the element boundary correction $\{\Delta U_f\}$. Here \bar{R} is the piecewise constant function with $\bar{R}|_{\Omega_e} = \bar{R}_e$. This is a linear least squares problem, the direct solution of which requires the solution of a globally coupled linear system.

To localize the computation, a series of corrections that are nonzero on only a single element boundary are defined

$$\Delta U_{f'} = c_f \delta_{ff'}, \quad 1 \leq f, f' \leq N_{f_i} \quad (58)$$

where c_f is chosen to minimize eqn (57), which leads to

$$c_f(R) = \frac{|\Omega_l|R_r - |\Omega_r|R_l}{|\gamma_f|(|\Omega_l| + |\Omega_r|)} \quad (59)$$

where Ω_l and Ω_r are the elements sharing face f , \mathbf{n}_f is directed from Ω_l to

Ω_r , and R is the piecewise constant function with $R|_{\Omega_e} = R_e$. After a face correction is computed, the affected element conservation residuals are updated in the manner of the classical Gauss-Seidel iteration for linear systems. The algorithm is not however equivalent to Gauss-Seidel for the global linear least squares problem.

To be concrete, the Sun-Wheeler iteration process is given in Algorithm 1. In Algorithm 1, k_m is the maximum number of iterations allowed, ϵ_{mc} is the mass conservation tolerance, and $f \in \mathcal{F}_i(e)$ corresponds to the set of interior element boundaries in $\partial\Omega_e$.

Algorithm 1 Sun-Wheeler Gauss-Seidel algorithm [18]

Require: $\bar{\sigma}_{h,f}$ that is globally conservative

- 1: $k = 0$. $\Delta U_f^{0,N_{f_i}} = 0$, $f = 1, \dots, N_f$. $R_e^{0,N_{f_i}}$ defined from eqn (56), $e = 1, \dots, N_e$.
 - 2: **while** $k < k_m$ and $\max_e |R_e^{k,N_{f_i}}| < \epsilon_{mc}$ **do**
 - 3: $R_e^{k,0} = R_e^{k-1,N_{f_i}}$, $e = 1, \dots, N_e$.
 - 4: $\Delta U_f^{k,0} = \Delta U_f^{k-1,N_{f_i}}$, $f = 1, \dots, N_{f_i}$.
 - 5: **for** $f = 1, \dots, N_{f_i}$ **do**
 - 6: $\Delta U_{f'}^{k,f} = \Delta U_{f'}^{k,f-1} + c_f(R^{k,f-1})\delta_{ff'}$, $f' = 1, \dots, N_{f_i}$.
 - 7: $R_e^{k,f} = R_e^{k,f-1} + \sum_{f' \in \mathcal{F}_i(e)} \int_{\gamma_{f'}} c_f(R^{k,f-1})\delta_{ff'} \mathbf{n}_{f'} \cdot \mathbf{n}_e ds$, $f' = 1, \dots, N_{f_i}$,
 $e = 1, \dots, N_e$.
 - 8: **end for**
 - 9: **end while**
-

Note that the original presentation of the algorithm in [18] was written in terms of \bar{R} and a different sign convention.

At the algorithm conclusion, we have a corrected velocity field defined on element boundaries

$$\hat{\sigma}_{h,f} = \bar{\sigma}_{h,f} + \Delta U_f \mathbf{n}_f \tag{60}$$

that can be extended to element interiors through the projection process onto local RT0 spaces as in eqn (54) above. In general, higher order mixed spaces can be used to define local Neumann problems to meet higher order compatibility conditions [18, 47].

A precondition for the Sun-Wheeler algorithm is a globally conservative velocity field defined on element boundaries, $\hat{\sigma}_{h,f}$. One established strategy for obtaining $\hat{\sigma}_{h,f}$ is an initial postprocessing that requires a projection over $\gamma_f \in \Gamma_D$ [22]. Another approach is to enforce Dirichlet boundary conditions weakly in the original finite element solution as in [25, 63]. We take the latter approach here. Specifically, when the Sun-Wheeler Gauss-Seidel postprocessing is used,

we modify the weak formulation in eqn (20) by reintroducing trial functions corresponding to Γ_D and adding the boundary integral

$$\int_{\Gamma_D} [(\mathbf{f} - \mathbf{a}\nabla\psi) \cdot \mathbf{n} + \sigma_p(\psi_h - \psi^b)] ds \quad (61)$$

where in this work $\sigma_p = 1/h$.

4.5 Solution methods

4.5.1 Nonlinear and linear solvers

The finite element method above leads to a discrete nonlinear system of equations that must be solved at each time step. These nonlinear problems are usually large for two and three-dimensional domains and difficult to solve. Variants of Picard iteration along with Newton or quasi-Newton methods have all been considered for Richards' equation [4, 5, 64]. However, no method has yet proven universally successful in our opinion.

The numerical experiments below are performed on a hierarchy of uniformly refined meshes, T_h^i , $i = 1, \dots, N_m$. Here, we use a standard Newton iteration with Armijo line search [65] on each level. For the steady-state examples, nested iteration (NLNI) [66, 67] is used to generate the initial Newton iterate on each level, to speed convergence. The resulting linear systems on each level are solved using the sparse direct SuperLU solver [68] for simplicity, since the focus of this work is the finite element formulation and velocity postprocessing. Other multilevel techniques and globalization strategies are certainly possible and can be advantageous in some cases [4, 67].

4.5.2 Local postprocessing systems

The postprocessing algorithm presented in §4.4.1 requires solution of N_n independent linear systems of size $N_{n,e}^* \times N_{n,e}^*$. To solve eqn (47), we use the LAPACK LU routines `dgetrf` and `dgetrs`. The node-star systems are refactored only when the mesh and/or Γ_N changes.

4.6 Time integration

For simplicity, we restrict the temporal approximation order to one in the transient test problem below, since the focus of this work is on the spatial

discretization and velocity postprocessing schemes. To control error, we require at each t^{n+1} that

$$\|m_h^{p,n+1} - m_h^{n+1}\|_\infty < \epsilon_r \|m_h^{n+1}\|_\infty + \epsilon_a \quad (62)$$

where $m_h^{p,n+1}$ is a first order predictor for the solution at t^{n+1} computed by extrapolating from m_h^n and m_h^{n-1} . A classical step-size controller is used for time step selection with maximum increase and decrease factors of two and one-tenth, respectively [69, p. 168]. For stabilized CG approximations, the evaluation of τ and ν_c is lagged a time step after an initial startup phase in an effort to simplify the nonlinear solves [70]. Δt is decreased by a factor of ten when a nonlinear solver failure is encountered.

5 Results

The finite element formulations given above admit many variations. In the following section, we present a series of numerical experiments designed to verify the basic methodology and investigate the relative performance of some of these variations. We consider a standard CG approximation as well as a multiscale-stabilized approach with shockcapturing (CG-S). Abbreviations for the methods used are given in Table 1.

For comparison, we also include results obtained with the P^1 nonconforming (NC) approach from [53] as discussed in §2.4. The NC approximation yields velocity fields equivalent to an RT0 approximation for the problems considered below with the advantage of a simple implementation within the same computational framework as the CG methods considered. The primary drawback is that a piecewise constant, average approximation for source terms and material coefficients is necessary for the nonconforming solution while this is not required for the CG solutions.

We combine the CG methods from Table 1 with different approaches for obtaining a velocity field, $\hat{\sigma}_h$. We denote the combined approach as (X-*) where X is one of CG, CG-S, or CG-V and the suffix depends on the evaluation of $\hat{\sigma}_h$. We allow for pointwise evaluation of $\hat{\sigma}_h$ via eqn (9) (PE) as well as the Larson-Niklasson postprocessing algorithm from §4.4.1 (LN) and the Sun-Wheeler algorithm from §4.4.2 (SW) with local RT0 representations.

Table 1
Summary for FEMs compared in numerical experiments

Abbrev.	Definition
CG	conforming Galerkin approximation, eqn (41) with $\tau = 0$, $\nu_c = 0$
CG-S	multiscale stabilized CG with shockcapturing, eqn (41), $\nu_c = 0.1^\dagger$
CG-V	lumped CG approximation with vertex quadrature, $\tau = 0$, $\nu_c = 0$
NC	P^1 nonconforming approach [53]

$\dagger \nu_c = 0.5$ for Problem V.

5.1 Test problems

Five example problems were selected for the numerical experiments. The majority are steady-state, since the emphasis of this work is on spatial approximation techniques. The first two problems are linear with a known analytical solution, which facilitates verification of the numerical model and evaluation of the methods' performance for cases where the solution is smooth. The third example is a steady-state recharge problem in a homogeneous domain and tests each method's ability to resolve fronts, since the solution contains an internal layer. The fourth problem consists of constant recharge in a block heterogeneous domain, while the fifth example considers transient infiltration into the same domain.

5.1.1 Problems I and II

The first two test problems are for steady-state, fully saturated flow in two (Problem I) and three-dimensional domains (Problem II). Sinusoidal and polynomial heterogeneity distributions are used in order to allow calculation of analytical solutions in two and three space dimensions, respectively. To be specific, we set $\Omega = [0, 1]^{n_d}$ and

$$\begin{aligned}\psi(\mathbf{x}) &= \sin^2(2\pi x_1) + \cos^2(2\pi x_2) + \sin^2(2\pi x_3) + x_1 + x_2 + x_3 + 5 \\ K_{s,ij}(\mathbf{x}) &= (5 + x_i^2)\delta_{ij}\end{aligned}\tag{63}$$

for $n_d = 2$ and

$$\begin{aligned}\psi(\mathbf{x}) &= \sum_{i=1}^3 x_i^2 \\ K_{s,ij}(\mathbf{x}) &= (5 + x_i^2 x_{i+1})\delta_{ij}\end{aligned}\tag{64}$$

for $n_d = 3$. In eqn (64) x_{i+1} is taken modulo 3. For simplicity, the gravity term is ignored. The source term, $b(\mathbf{x})$, and boundary conditions are calculated from eqn (1) assuming eqns (63) and (64). For Problem II, Dirichlet conditions are set everywhere except the face $x_3 = 0$. Dirichlet conditions are set everywhere except the segment $x_2 = 1$, for Problem I.

5.1.2 Problem III

The third example involves steady-state variably saturated groundwater flow assuming constant recharge, $\sigma^b = -2 \times 10^{-2}$ [m/d] into a two-dimensional homogeneous domain, $\Omega = [0, 1] \times [0, 5]$ [m²]. The medium corresponds to the ‘‘Sand’’ test problem used in [71, see Table 1] as well as [3, 7]. To be concrete, VGM p - s - k relations are used with $n_{vg} = 4.26$, and $\alpha_{vg} = 5.47$, while $\theta_s = 0.301$, $\theta_r = 0.093$, and the saturated conductivity is isotropic with $K_s = 5.04$ [m/d]. We neglect compressibility of the aqueous phase, $\beta_c = 0$ for simplicity, however.

A fixed pressure head, $\psi = 1$ [m] is specified on the domain bottom and no flow along the vertical boundaries. A constant value of $\psi = 1$ [m] is also used as the initial guess for the Newton solve. The resulting solution contains a steep transition zone and a constant velocity $\boldsymbol{\sigma} = [0, \sigma^b]^T$ everywhere. Even though the solution is essentially one-dimensional, it provides a reasonable test for the ability of the finite element methods to resolve the boundary layer and maintain symmetry in the velocity field.

5.1.3 Problems IV and V

The fourth and fifth examples are based on the second test problem from [10]. VGM p - s - k relations are again used with four separate media types configured in a simple block pattern. The domain consists of two shallow layers near the surface and a third, larger region with a small high conductivity sub-block. The media properties and domains are summarized in Table 2. To be consistent with [10], we set $\beta_c = 0$ as well. The physical domain is $\Omega = [0, 8] \times [0, 6.5]$ [m²].

Problem IV is a steady-state example. A constant recharge of $\sigma^b = -2 \times 10^{-2}$ [m/d] is set on the top left 2.25 meters of the domain. No flow boundaries are set along the remainder of the top boundary, as well as the left boundary. Hydrostatic conditions are set on the right boundary with a water table elevation of 2.17 [m].

Problem V considers transient infiltration and has no flow boundary conditions along the right boundary rather than fixed ψ . We set the initial condition to $\psi^0 = -89.96$ [m].

Table 2

Media properties for Problems IV and V

Type	K_s [m/d]	θ_s [-]	θ_r [-]	α_v [1/m]	n_v [-]	Location
1	7.89	0.3680	0.1020	3.334	1.982	$[0, 8] \times [6.0, 6.5]$
2	4.69	0.3510	0.09849	3.63	1.632	$[0, 8] \times [5.5, 6.0]$
3	4.143	0.3250	0.08590	3.455	5	Ω^\dagger
4	41.143	0.3250	0.08590	3.455	5	$[1, 3] \times [4, 5]$

† medium type is 3 by default

5.2 Error and work measures

To measure accuracy of numerical approximations for ψ and $\boldsymbol{\sigma}$, we use relative error in discrete L^p norms

$$\varepsilon_{u,p} = \frac{\|u_h - \tilde{u}\|_p}{\|\tilde{u}\|_p} \quad (65)$$

where u is generic variable placeholder and $p = 1, 2$, or ∞ . \tilde{u} is the analytical solution when available or a reference numerical solution otherwise. When analytical solutions are not available, the discrete solution on the finest mesh in the multilevel hierarchy is used as the reference solution and coarser solutions are projected onto the finest mesh in order to approximate error. This was judged to give an acceptable measure of solution quality even though error in the reference solution is nonnegligible for Problems IV and V given their heterogeneous, nonlinear character [11].

Mass conservation error is recorded as

$$\varepsilon_{mc} = \max_{\Omega_e} \left| \int_{\Omega_e} \hat{m}_t dx + \int_{\partial\Omega_e} \hat{\boldsymbol{\sigma}}_h \cdot \mathbf{n}_e ds - \int_{\Omega_e} b dx \right| \quad (66)$$

where $\hat{\boldsymbol{\sigma}}_h$ is a velocity field obtained from one of the postprocessing methods described above: PE, LN, SW, or the NC approximation.

There are several ways to measure computational effort, including number of degrees freedom, N_{dof} as well as total CPU time [72]. We rely primarily on total degrees of freedom here, since the majority of the computations for the methods compared are implemented within the same general-purpose finite element library. The sparsity pattern of the NC approximation can affect performance [73] in general, but the impact on SuperLU is minimal beyond the difference in total degrees of freedom. Since the LN postprocessing algorithm

requires building and factoring local systems for each $\mathcal{E}(n)$, we report the approximate CPU overhead for each mesh as well, where the CPU times are averaged over at least 1000 repetitions. For the SW algorithm, we report the CPU time required to reach convergence or exhaust the allowed number of iterations for each simulation.

5.3 Implementation details

Element integrals in Problem I and II were approximated using Gaussian quadrature that is exact for quartics and cubics, respectively. Fourth-order Gaussian quadrature was again used for the CG approximations in Problem III, while only third-order quadrature was used for the nonlinear, heterogeneous examples. Second-order quadrature with points taken from element boundary barycenters was used for the NC approximations in the nonlinear examples. The nonlinear solver employed a relative residual convergence test. For Problem III, absolute and relative tolerances of $\epsilon_{nl,a} = 1 \times 10^{-8}$ and $\epsilon_{nl,r} = h \times 10^{-3}$, where h is the mesh diameter were used. For Problems IV and V, $\epsilon_{nl,a} = 1 \times 10^{-6}$, while $\epsilon_{nl,r} = h \times 10^{-6}$. The absolute and relative temporal integration tolerances were set to $\epsilon_a = \epsilon_r = 10^{-3}$ for Problem V. A convergence tolerance of $\epsilon_{mc} = 10^{-6}$ and was used for the SW postprocessing in all cases and the maximum number of iterations was set $k_{max} = 10000$.

The computations were performed using a finite element library under development by researchers at the US Army Corps of Engineers, Clemson University, North Carolina State University, the University of Texas at Austin, and Applied Research Associates, Inc. High level routines and methods were implemented in Python, while low-level computationally intensive portions were in c and F77. Simulations were run on a 3 GHz duo core Intel Xeon Mac Pro desktop with 8 GB memory. Python 2.5, gcc 4.0.1, and g77 3.4.0 were used to interpret and compile the code.

5.4 Simulation Results

5.4.1 Problems I and II

Tables 3 and 5 record the accuracy of the CG and NC approximations for ψ on uniformly refined meshes for Problem I and Problem II, respectively, while Tables 4 and 6 compare accuracy and local mass conservation for the different velocity postprocessing techniques. The CPU overhead for building and factoring the local node-star systems, eqn (46), on each mesh are reported in Table 7.

Stabilization and vertex quadrature were not considered for Problems I and II, since the solutions were smooth. Looking at Tables 3 and 5, both CG and NC methods obtained second-order convergence for ψ , as expected. For Problem I, the accuracy of the NC solution was clearly limited by the use of piecewise constant average values for b , since the CG and NC error values were nearly identical, even though N_{dof} for the NC approximation was significantly higher. For Problem II, the NC errors were significantly lower for a given level of refinement, but the CG accuracy was slightly better per degree of freedom.

Global accuracy of the velocity postprocessing methods was similar for Problem I. In particular, point-wise evaluation was first-order accurate even though local mass conservation failed. For Problem II the CG-PE approximation was again first order accurate, but the actual error values were two to three times higher. The accuracy of the LN and SW velocity fields was essentially the same. On each level, the NC approximation error for $\hat{\sigma}_h$ was significantly lower than that of the LN and SW approximations, but the postprocessing algorithms accuracy was competitive if not better in terms of accuracy per degree of freedom. The costs in CPU time clearly increased for $n_d = 3$ due to the increased nodal and element boundary connectivity.

Table 3
 $\varepsilon_{\psi,2}$, Problem I

Method	Level	h	N_{dof}	$\varepsilon_{\psi,2}$	Rate
CG	1	0.354	25	1.23×10^{-2}	-
NC	1	0.354	56	5.76×10^{-2}	-
CG	2	0.177	81	1.520×10^{-2}	-0.225
NC	2	0.177	208	1.51×10^{-2}	1.85
CG	3	0.0884	289	3.97×10^{-3}	1.93
NC	3	0.0884	800	4.14×10^{-3}	1.94
CG	4	0.0442	1089	1.01×10^{-3}	1.98
NC	4	0.0442	3136	1.05×10^{-3}	1.98
CG	5	0.0221	4225	2.52×10^{-4}	2.00
NC	5	0.0221	12416	2.64×10^{-4}	1.99

5.4.2 Problem III

Figure 3 compares approximate solutions for m from Problem III after uniformly refining an initial eight element mesh four times. The resulting mesh was still fairly coarse, with 512 elements, 800 edges, and 289 nodes. For simplicity, the NC approximation is shown using nodal values obtained by averaging the corresponding vertex values from neighboring elements. Accuracy of the

Table 4
 $\varepsilon_{\sigma,2}$ and ε_{mc} , Problem I

Method	Level	$\varepsilon_{\sigma,2}$	rate	ε_{mc}^\dagger
CG-PE	1	4.44×10^{-1}	-	1.30×10^1
CG-LN	1	7.60×10^{-1}	-	0
CG-SW	1	8.30×10^{-1}	-	9.94×10^{-7}
NC	1	6.98×10^{-1}	-	0
CG-PE	2	4.25×10^{-1}	-8.84×10^{-1}	6.54×10^0
CG-LN	2	4.21×10^{-1}	6.32×10^{-1}	0
CG-SW	2	4.56×10^{-1}	8.63×10^{-1}	9.94×10^{-7}
NC	2	4.16×10^{-1}	7.45×10^{-1}	0
CG-PE	3	2.19×10^{-1}	9.58×10^{-1}	2.07×10^0
CG-LN	3	2.19×10^{-1}	9.39×10^{-1}	0
CG-SW	3	2.22×10^{-1}	1.04×10^0	9.94×10^{-7}
NC	3	2.17×10^{-1}	9.37×10^{-1}	0
CG-PE	4	1.11×10^{-1}	9.89×10^{-1}	6.28×10^{-1}
CG-LN	4	1.10×10^{-1}	9.84×10^{-1}	0
CG-SW	4	1.11×10^{-1}	1.00×10^0	9.99×10^{-1}
NC	4	1.10×10^{-1}	9.81×10^{-1}	0
CG-PE	5	5.54×10^{-2}	9.97×10^{-1}	2.05×10^{-1}
CG-LN	5	5.53×10^{-2}	9.96×10^{-1}	0
CG-SW	5	5.54×10^{-2}	1.00×10^0	1.0×10^{-6}
NC	5	5.53×10^{-2}	9.95×10^{-1}	0

† 0 means $< 1 \times 10^{-12}$

CG and NC approximations is summarized in Table 8, where $\varepsilon_{\sigma_1,\infty}$ and $\varepsilon_{\sigma_2,\infty}$ are discrete L_∞ errors for the transverse and longitudinal velocity coordinates, respectively. $\varepsilon_{\psi,\infty}$ was estimated using the a reference solution obtained from seven levels of refinement ($N_{dof} = 16641$) in the case of the CG methods, and six levels of refinement for the NC solution ($N_{dof} = 12416$). Table 9 gives the CPU effort for the velocity postprocessing schemes.

From Table 8, it is clear that the LN technique conserved mass up to the non-linear solver error, regardless of the CG variation with which it was combined. Similarly, the SW algorithm conserved mass up to the imposed conservation tolerance, $\varepsilon_{mc} = 10^{-6}$. The NC velocity approximation was essentially exact

Table 5
 $\varepsilon_{\psi,2}$, Problem II

Method	Level	h	N_{dof}	$\varepsilon_{\psi,2}$	Rate
CG	1	0.433	125	2.75×10^{-2}	-
NC	1	0.433	864	3.63×10^{-3}	-
CG	2	0.217	729	6.87×10^{-3}	2.00
NC	2	0.217	6528	8.85×10^{-4}	2.05
CG	3	0.108	4913	1.72×10^{-3}	2.00
NC	3	0.108	50688	2.17×10^{-4}	2.03

Table 6
 $\varepsilon_{\sigma,2}$ and ε_{mc} , Problem II

Method	Level	$\varepsilon_{\sigma,2}$	rate	ε_{mc}^\dagger
CG-PE	1	1.22×10^{-1}	-	2.87×10^{-1}
CG-LN	1	6.38×10^{-2}	-	0
CG-SW	1	5.80×10^{-2}	-	9.90×10^{-7}
NC	1	1.29×10^{-2}	-	0
CG-PE	2	6.09×10^{-2}	1.01×10^0	8.27×10^{-2}
CG-LN	2	2.74×10^{-2}	1.22×10^0	0
CG-SW	2	2.95×10^{-2}	9.75×10^{-1}	9.97×10^{-7}
NC	2	6.69×10^{-3}	9.50×10^{-1}	0
CG-PE	3	3.04×10^{-2}	1.00×10^0	2.21×10^{-2}
CG-LN	3	1.19×10^{-2}	1.20×10^0	0
CG-SW	3	1.39×10^{-2}	1.09×10^0	1.00×10^{-6}
NC	3	3.39×10^{-3}	9.82×10^{-1}	0

† 0 means $< 1 \times 10^{-12}$

up to the nonlinear solver tolerance, while the error in both postprocessing schemes was significantly lower than the error in the directly evaluated velocity field (CG-PE). The LN approximations were more accurate than SW solutions on the level four mesh, and the CG-V velocities were more accurate than those obtained from the CG-S approximation, apparently due to excessive numerical diffusion from the shock capturing term.

Comparing error values for ψ and the solution profiles from Figure 3, $\varepsilon_{\psi,\infty}$ corresponds to undershoot at the front. The lumped (CG-V) and consistent (CG) approximations incur the most undershoot. The multiscale stabilized

Table 7

CPU overhead for velocity postprocessing algorithms, Problems I and II

Problem	Level	N_n	N_f	LN CPU [†] [s]	SW CPU* [s]
I	1	25	56	0	0 (83)
I	2	81	208	0	1.67×10^{-2} (342)
I	3	289	800	2.15×10^{-3}	1.17×10^{-1} (907)
I	4	1089	3136	6.47×10^{-3}	9.67×10^{-1} (1687)
I	5	4225	12416	3.22×10^{-2}	1.78×10^0 (831)
II	1	125	864	2.41×10^{-3}	3.33×10^{-2} (243)
II	2	729	6528	2.45×10^{-2}	3.83×10^{-1} (466)
II	3	4913	50688	2.65×10^{-1}	4.23×10^0 (484)

† 0 means $< 1 \times 10^{-3}$ [s]

* number of iterations in parentheses

approach with shock-capturing was the only method able to resolve the solution monotonically on the coarser meshes. As one would expect, the relative advantage of the CG-S approach decreased as the mesh was refined and the other methods were then able to resolve the internal layer. For instance, the NC solution did not fully avoid undershoot until the fifth level of refinement although the nodal averages shown in 3 are clearly more accurate than the CG-S solution.

Table 8

Problem III solution error

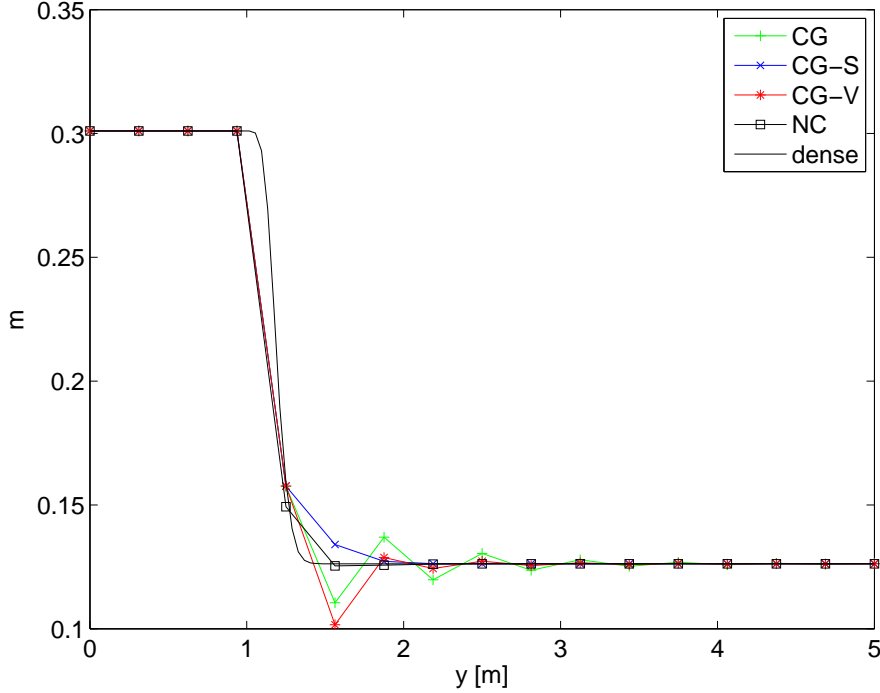
Method	Level [†]	N_{dof}	$\varepsilon_{\psi, \infty}$	ε_{mc}	$\varepsilon_{\sigma_1, \infty}$	$\varepsilon_{\sigma_2, \infty}$
CG-PE	4	289	8.72×10^{-2}	2.26×10^{-3}	4.40×10^{-3}	2.19×10^0
CG-LN	4	289	8.72×10^{-2}	1.35×10^{-8}	4.37×10^{-4}	1.89×10^{-1}
CG-S-LN	4	289	4.36×10^{-2}	3.00×10^{-7}	3.90×10^{-4}	3.65×10^{-1}
CG-S-SW	4	289	4.36×10^{-2}	9.97×10^{-7}	2.10×10^{-3}	6.91×10^{-1}
CG-V-LN	4	289	1.75×10^{-1}	9.33×10^{-9}	2.29×10^{-4}	9.84×10^{-2}
CG-V-SW	4	289	1.75×10^{-1}	9.93×10^{-7}	6.99×10^{-4}	2.49×10^{-1}
NC	4	800	9.49×10^{-2}	0	5.21×10^{-8}	1.66×10^{-4}
CG-PE	5	1089	3.28×10^{-2}	7.17×10^{-4}	2.22×10^{-3}	1.61×10^0
CG-LN	5	1089	3.28×10^{-2}	6.10×10^{-8}	3.29×10^{-4}	1.71×10^{-1}
CG-S-LN	5	1089	2.30×10^{-2}	8.88×10^{-8}	3.39×10^{-4}	3.58×10^{-1}
CG-S-SW	5	1089	2.30×10^{-2}	9.97×10^{-7}	1.11×10^{-3}	3.98×10^{-1}
CG-V-LN	5	1089	5.58×10^{-2}	4.92×10^{-9}	2.54×10^{-4}	1.12×10^{-1}
CG-V-SW	5	1089	5.58×10^{-2}	9.98×10^{-7}	4.99×10^{-4}	1.45×10^{-1}
NC	5	3136	2.32×10^{-2}	0	2.71×10^{-7}	1.55×10^{-5}

† $h = 0.319$ on level 4, and $h = 0.160$ on level 5

Table 9

CPU overhead for velocity postprocessing algorithms, Problem III

Method	Level	Its.	CPU [s]
LN	4	-	1.35×10^{-3}
CG-V-SW	4	250	3.33×10^{-2}
CG-S-SW	4	595	6.67×10^{-2}
LN	5	-	5.52×10^{-3}
CG-V-SW	5	83	3.33×10^{-2}
CG-S-SW	5	207	1.17×10^{-1}

Fig. 3. Profile of m along $x = 0.5[m]$ on mesh level 4

5.4.3 Problem IV

Figure 4 illustrates the problem domain and initial mesh generated by Triangle [74] for Problem IV and V. Performance of the methods for Problem IV is summarized in Table 10. CPU overhead for the LN and SW algorithms is shown in Table 11. NLNI failed with the CG approximation for Problem IV, due to failure in the Newton solve on coarser levels in the mesh hierarchy. Instead, the CG results in Table 10 were obtained by solving Newton's method independently on levels 3,4, and 5, and we also provide the ratio of the CPU time for the LN and SW algorithms to the CPU time for Newton's method as a percentage, which demonstrates the minor cost of the post-processing approaches relative to the cost of calculating the finite element solution.

Figure 5 compares the LN $\hat{\sigma}_h$ obtained from the CG-S approximation and the NC solution for $\hat{\sigma}_h$ after three levels of refinement. The CG-V-LN solutions for ψ and $\hat{\sigma}_h$ on level five are shown in Figure 6. The velocity fields in Figure 5 are similar and close to the CG-V-LN solution on five levels of refinement except for a deviation in the CG-S-LN solution around (4.5,2.5). This was generally true of the various method combinations listed in Table 10 with the exception of the point-wise evaluated $\hat{\sigma}_h$, and the CG-S-SW $\hat{\sigma}_h$ on level three which had nearly an order of magnitude greater error than the remaining approximations.

Both postprocessing schemes maintained local mass conservation, but the relative expense of the SW approach increased for Problem IV. This was particularly evident with the stabilized approximations (CG-S), where between two and four times as many iterations were required than with the lumped approximation. Accuracy in terms of $\varepsilon_{\psi,2}$ was comparable among the methods, given the level of error likely in the finest level approximation.

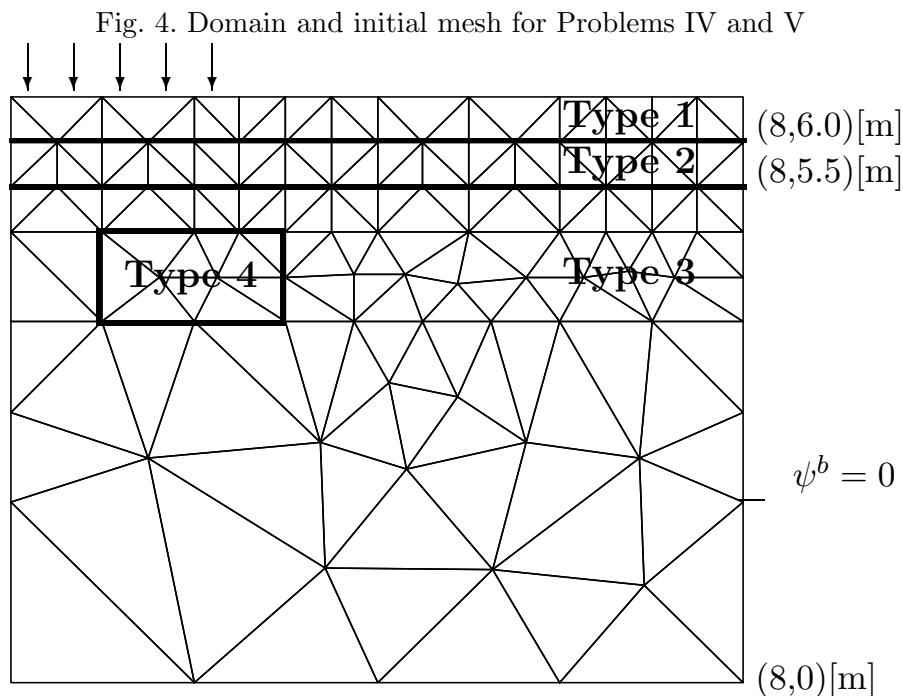


Table 10
Solution errors, Problem IV

Method	Level [†]	N_{dof}	ε_{mc}	$\varepsilon_{\psi,2}$	$\varepsilon_{\sigma,2}$
CG-LN	3	1351	3.55×10^{-11}	2.37×10^{-2}	1.20×10^{-1}
CG-V-LN	3	1351	1.20×10^{-8}	3.64×10^{-2}	1.22×10^{-1}
CG-V-SW	3	1351	1.00×10^{-6}	3.64×10^{-2}	1.83×10^{-1}
CG-S-PE	3	1351	1.86×10^{-1}	6.35×10^{-2}	1.94×10^0
CG-S-LN	3	1351	1.08×10^{-6}	6.35×10^{-2}	1.83×10^{-1}
CG-S-SW	3	1351	9.98×10^{-7}	6.35×10^{-2}	1.18×10^0
NC	3	3926	0	1.44×10^{-2}	1.18×10^{-1}
CG-LN	4	5277	2.71×10^{-11}	9.45×10^{-3}	5.35×10^{-2}
CG-V-LN	4	5277	8.00×10^{-9}	2.08×10^{-2}	5.37×10^{-2}
CG-V-SW	4	5277	1.00×10^{-7}	2.08×10^{-2}	8.56×10^{-2}
CG-S-PE	4	5277	2.45×10^{-2}	8.59×10^{-3}	2.12×10^{-1}
CG-S-LN	4	5277	6.67×10^{-7}	8.59×10^{-3}	5.72×10^{-2}
CG-S-SW	4	5277	1.00×10^{-6}	8.59×10^{-3}	1.01×10^{-1}
NC	4	15580	0	6.27×10^{-3}	5.24×10^{-2}

[†] $h = 0.707$ on level 3, $h = 0.354$ on level 4, and $h = 0.177$ on level 5

Table 11
CPU overhead for LN and SW algorithms, Problem IV

Method	Level	Its.	CPU [s]	% of Newton CPU
LN	3	-	7.14×10^{-3}	0.0734
CG-V-SW	3	651	4.67×10^{-1}	4.80
CG-S-SW	3	1714	1.20×10^0	12.3
LN	4	-	3.45×10^{-2}	0.106
CG-V-SW	4	586	1.52×10^0	4.69
CG-S-SW	4	1195	3.12×10^0	9.63
LN	5	-	1.56×10^{-1}	0.274
CG-V-SW	5	140	1.60×10^0	2.81
CG-S-SW	5	575	8.15×10^0	14.3

Fig. 5. $\hat{\sigma}_h$ CG-S-LN level 3 (left), NC (right)

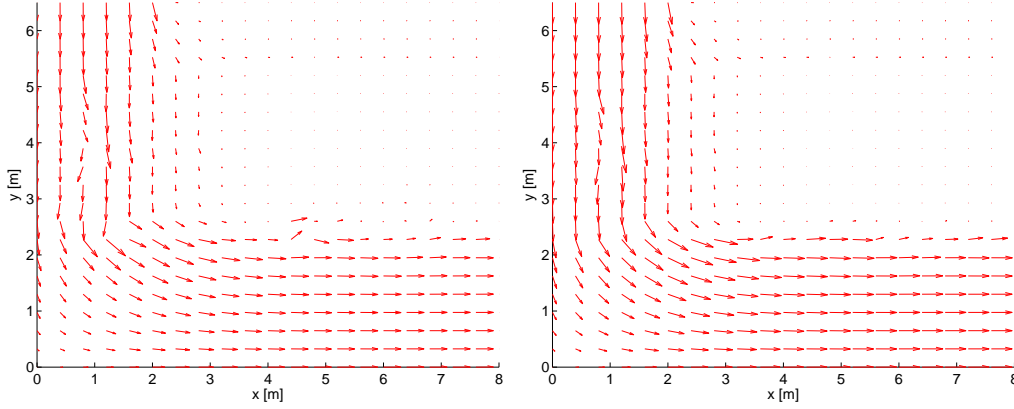
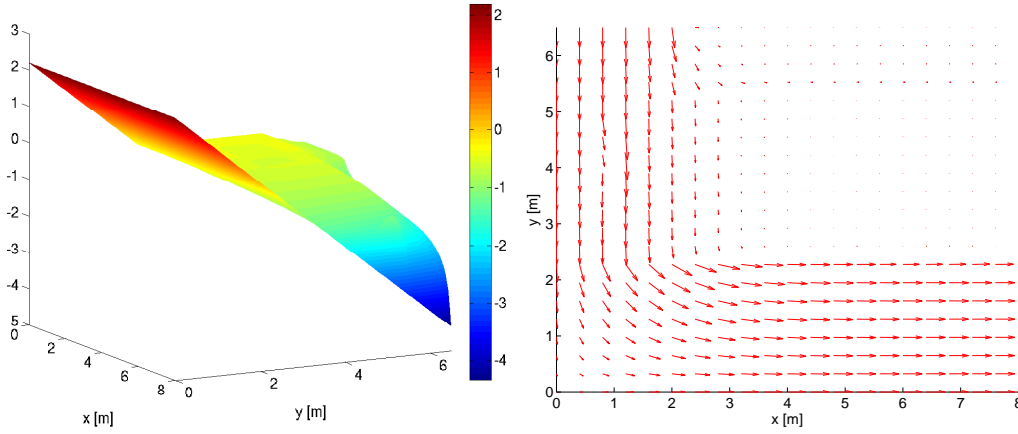


Fig. 6. CG-V level 5 ψ (left), $\hat{\sigma}_h$ (right)



5.4.4 Problem V

Problems IV and V differed only in the boundary condition on the right face of the domain and the transient nature of Problem V. Since Problem IV demonstrated the accuracy of the postprocessing schemes for heterogeneous problems, the simulations for Problem V focused on the ability of the CG-S and CG-V methods to resolve the infiltration front monotonically and the overall expense of the LN and SW schemes when used at each time step of a transient simulation. Table 12 shows mass conservation error and computational effort for the LN and SW algorithms when combined with the CG-S and CG-V discretizations. Figures 8 and 9 compare the solution fronts at $T = 30$ [d] for CG-S and CG-V on three levels of refinement. Figure 9 shows the CG-V solution for ψ on five levels of refinement. Upwinding was not necessary for the CG-V approximation for these refinement levels, and both solutions had little or no overshoot. Although we did not do a detailed comparison of the NC solution to the CG-S and CG-V solutions for this problem, we did find that the NC solution had significant undershoot.

The three-level CG-V solution smeared the infiltration front enough to almost

reach the lower right corner of the physical domain at $T = 30$ [d]. On the other hand, the CG-S solution smeared the infiltration front out significantly in the horizontal direction, so that the leading edge of the front lagged the CG-V solution along the bottom no-flow boundary. The lateral smearing of the CG-S solution was caused primarily by the isotropic nature of the shock-capturing diffusion and the relatively large value $\nu_c = 0.5$ necessary to obtain an oscillation-free solution.

The CPU times reported in Table 12 include the initial expense required for building and factoring the local node-star systems as well as the per iteration backsubstitutions. Since the mesh and boundary conditions did not change for Problem V, the overall computational advantage for the LN postprocessing increased significantly. In addition, we note that the SW algorithm failed to converge within the maximum allowed iterations for two time steps on level four and failed for four time steps on level five. However, the maximum mass conservation error in these cases was 3.19×10^{-5} .

Table 12
LN and SW performance, Problem V

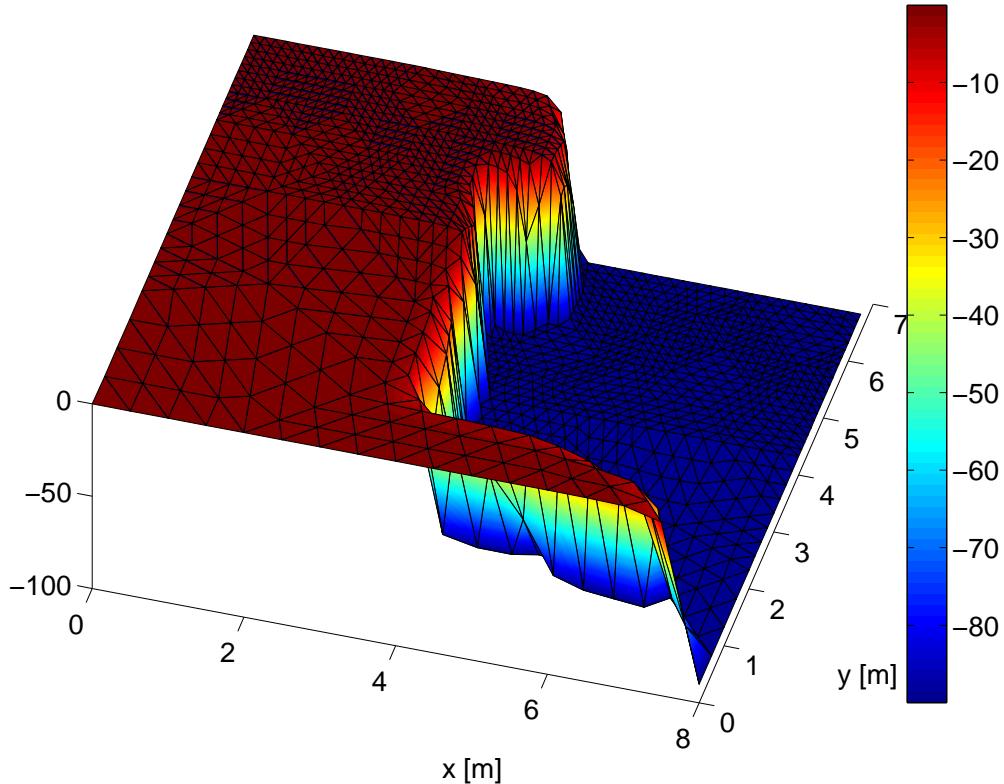
Method	Level	ε_{mc} at $T = 30[d]$	Avg. its.	Avg. CPU [†] [s]
CG-V-LN	3	1.18×10^{-7}	-	1.52×10^{-3} (5.62×10^{-3})
CG-V-SW	3	9.97×10^{-7}	451	3.13×10^{-1}
CG-S-LN	3	3.61×10^{-7}	-	1.52×10^{-3} (5.62×10^{-3})
CG-S-SW	3	1.00×10^{-6}	2289	1.58×10^0
CG-V-LN	4	2.09×10^{-7}	-	9.70×10^{-3} (2.49×10^{-2})
CG-V-SW	4	9.99×10^{-7}	377	9.74×10^{-1}
CG-S-LN	4	4.73×10^{-8}	-	9.70×10^{-3} (2.49×10^{-2})
CG-S-SW	4	1.00×10^{-6}	2072	5.39×10^0
CG-V-LN	5	5.37×10^{-7}	-	4.17×10^{-2} (1.14×10^{-1})
CG-V-SW	5	9.97×10^{-7}	185	2.13×10^0
CG-S-LN	5	1.52×10^{-7}	-	4.17×10^{-2} (1.14×10^{-1})
CG-S-SW	5	1.00×10^{-6}	584	8.20×10^0

† CPU required to build and factor LN node-star systems in parentheses

5.5 Discussion

There were two basic aims of this work. The first was to formulate a multiscale, stabilized finite element approximation for Richards' equation. The

Fig. 7. ψ CG-V level 3 solution

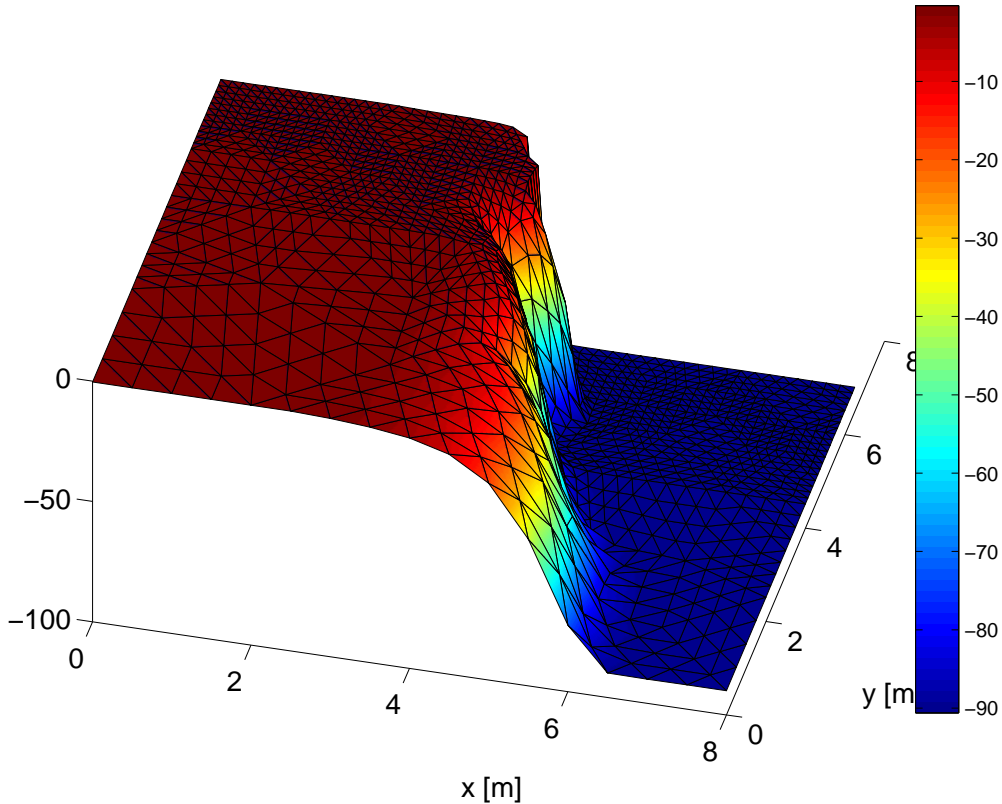


second was to detail two postprocessing techniques that are capable of producing locally conservative velocity fields from traditional and stabilized finite element solutions for pressure-head. The goal of the numerical experiments was to determine if the multiscale-stabilized strategy could provide improved approximations for ψ and to evaluate the accuracy and computational efficiency of the velocity postprocessing algorithms.

The numerical experiments covered a range of conditions from steady-state single-phase flow to variably saturated, block heterogeneous infiltration problems. The multiscale stabilization was compared to a traditional conforming finite element approximation with and without mass-lumping, while the impact of velocity postprocessing was measured by comparison to direct evaluation of σ via Darcy's law. As a point of reference, we also considered a locally conservative nonconforming finite element approximation that coincides with a mixed hybrid finite element approximations in many cases.

The velocity postprocessing algorithm from Larson and Niklasson [17] combined with a local RT0 representation worked well with each of the conforming Galerkin methods considered. Local mass conservation was always obtained, and global accuracy was improved. The postprocessing algorithm applied to piecewise linear discretizations produces linear normal fluxes on element boundaries, which naturally matches a linear BDM1 or BDDF1 rep-

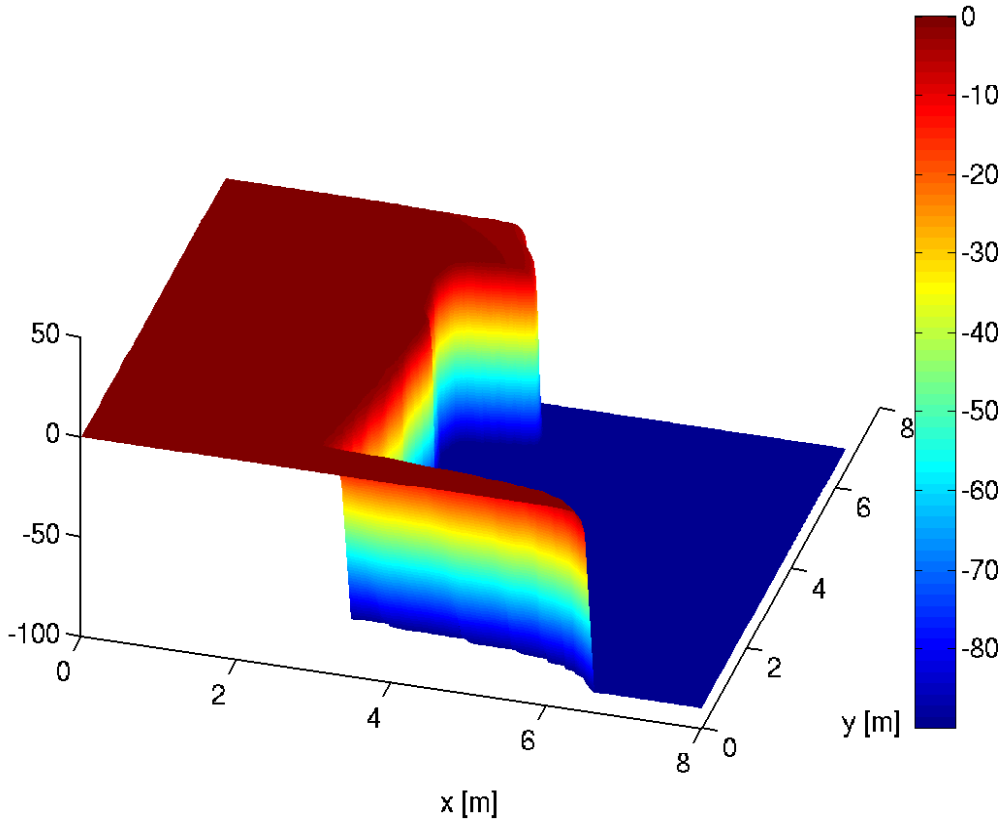
Fig. 8. ψ CG-S level 3 solution



representation on element interiors. While not shown, we found that use of a local BDM1 or BDDF1 representation improved accuracy for some cases such as Problem I. In general, the improvements were not sufficient to justify the added storage ($n_d^2 + n_d$ versus $n_d + 1$ local degrees of freedom), since the post-processed boundary fluxes were only first order accurate [17]. On the other hand, local projections onto higher-order mixed spaces following the approach laid out in [18] could still be used to obtain velocity fields satisfying higher order compatibility conditions [47]. The algorithm is theoretically scalable for parallel implementations.

The Sun-Wheeler postprocessing scheme produced mass-conservative velocity fields with accuracy similar to that of the Larson and Niklasson algorithm, with the exception of coarse grid, multiscale-stabilized solutions to Problem IV. The Sun-Wheeler algorithm usually required more CPU time than the Larson-Niklasson approach, and it failed to converge in the maximum number of iterations allowed for a few (four) time steps in Problem V. The CPU time required for both methods was a small fraction of the total solver time in most cases, however, and the mass conservation residual was still small (at least 3.18×10^{-5}) when the Sun-Wheeler iterations failed to converge. The Sun-Wheeler algorithm as presented is inherently sequential with a convergence rate depending on the mesh, but implementation is quite simple.

Fig. 9. ψ CG-V level 5 solution



We have tried to be comprehensive in the numerical experiments, yet there are several factors that were not accounted for directly. For instance, the Sun-Wheeler algorithm requires minimal storage, while the Larson-Niklasson approach required storage of the local, factored node-star systems in our implementation. This memory overhead can be non-trivial for large, three-dimensional problems. Similarly, we did not consider local mesh refinement, which would have required rebuilding some of the Larson-Niklasson node-star systems and likely increased its overhead. We also did not include the global algorithms from [17, 18] which might be competitive with a sufficiently fast linear solver [18].

The RT0 velocity fields from the velocity postprocessing schemes were typically comparable in accuracy to RT0 velocity fields obtained from the non-conforming discretization, and were often cheaper to obtain in terms of total degrees of freedom. While these results do not attempt to provide a comprehensive evaluation of the relative merits of primal conforming and mixed finite element approximations, they do indicate that the proposed approach should be at least competitive with mixed methods for Richards' equation.

The effectiveness of the multiscale stabilization strategy varied somewhat. For a steady-state, variably saturated example where the solution contained a

sharp internal layer (Problem III), it improved resolution over the other methods considered on coarse grids. On the other hand, its advantages over traditional mass-lumping were more limited for Problem V. Although the framework used here is quite general, the stabilization and shock capturing parameters were evaluated using direct extensions of standard formulas for nonlinear advection-diffusion-reaction problems, and the discrete strong residual was approximated using the chain-rule for simplicity. These approaches clearly need refinement to obtain more robust and accurate approximations for Richards' equation.

Acknowledgements

Special thanks to John Crispell, Lea Jenkins, Scott Pope, Steven Antrim, and Wayne Tanner for contributing source code to the finite element library, and to Stacy Howington, Charlie Berger, and Tim Kelley for participating in discussions on the numerical methods. Chris Kees was partially supported by a postdoctoral fellowship at the Center for Research in Computation in the Department of Mathematics at North Carolina State University and the J. Tinsley Oden visiting faculty program at the Institute for Computational Engineering and Sciences. Clint Dawson was supported by the National Science Foundation grant DMS 0411413. Permission was granted by the Chief of Engineers to publish this information.

References

- [1] G. de Marsily, *Quantitative Hydrogeology: Groundwater Hydrology for Engineers*, Academic Press, San Diego, CA, 1986.
- [2] T. P. Witelski, Horizontal infiltration into wet soil, *Water Resources Research* 34 (7) (1998) 1859–1863.
- [3] H. Li, M. W. Farthing, C. N. Dawson, C. T. Miller, Local discontinuous Galerkin approximations to Richards' equation, *Advances in Water Resources* 30 (2007) 555–575.
- [4] M. W. Farthing, C. E. Kees, T. S. Coffey, C. T. Kelley, C. T. Miller, Efficient steady-state solution techniques for variably saturated groundwater flow, *Advances in Water Resources* 26 (8) (2003) 833–849.
- [5] C. Paniconi, M. Putti, A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, *Water Resources Research* 30 (12) (1994) 3357–3374.
- [6] M. A. Celia, E. T. Bouloutas, R. L. Zarba, A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resources Research* 26 (7) (1990) 1483–1496.

- [7] C. E. Kees, C. T. Miller, Higher order time integration methods for two-phase flow, *Advances in Water Resources* 25 (2) (2002) 159–177.
- [8] G. Manzini, S. Ferraris, Mass-conservative finite volume methods on 2-D unstructured grids for the Richards’ equation, *Advances in Water Resources* 12 (2004) 1199–1215.
- [9] A. Ern, J.-L. Guermond, *Theory and Practice of Finite Elements*, No. 159 in *Applied Mathematical Sciences*, Springer-Verlag, New York, New York, 2004.
- [10] P. Forsyth, M. Kropinski, Monotonicity considerations for saturated-unsaturated subsurface flow, *SIAM Journal on Scientific Computing* 18 (5) (1997) 1328–1354.
- [11] M. W. Farthing, C. E. Kees, C. T. Miller, Mixed finite element methods and higher order temporal approximations for variably saturated groundwater flow, *Advances in Water Resources* 26 (4) (2003) 373–394.
- [12] M. Bause, P. Knabner, Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods, *Advances in Water Resources* 27 (2004) 565–581.
- [13] R. Srivastava, T. C. J. Yeh, A 3-Dimensional numerical model for water flow and transport of chemically reactive solute through porous media under variably saturated conditions, *Advances in Water Resources* 15 (5) (1992) 275–287.
- [14] C. S. Woodward, C. N. Dawson, Analysis of expanded mixed finite element methods for a nonlinear parabolic equation modeling flow into variably saturated porous media, *SIAM Journal on Numerical Analysis* 37 (3) (2000) 701–724.
- [15] C. T. Miller, C. Abhishek, A. B. Sallerson, J. F. Prins, M. W. Farthing, A comparison of computational and algorithmic advances for solving Richards’ equation, in: C. T. Miller, M. W. Farthing, W. G. Gray, G. F. Pinder (Eds.), *Proceedings of the XVth International Conference on Computational Methods in Water Resources*, Elsevier, Chapel Hill, North Carolina, 2004, pp. 1131–1145.
- [16] T. Hughes, G. Feijóo, L. Mazzei, J. Quincy, The variational multiscale method — a paradigm for computational mechanics, *Computer methods in Applied Mechanics and engineering* 166 (1998) 3–24.
- [17] M. Larson, A. Niklasson, A conservative flux for the continuous Galerkin method based on discontinuous enrichment, *CALCOLO* 41 (2004) 65–76.
- [18] S. Y. Sun, M. F. Wheeler, A posteriori error estimation and dynamic adaptivity for symmetric discontinuous Galerkin approximations of reactive transport problems, *Computer Methods in Applied Mechanics and Engineering* 195 (7-8) (2006) 632–652.
- [19] T. Hughes, G. Wells, Conservation properties for the Galerkin and stabilised forms of the advection-diffusion and incompressible Navier-Stokes equations, *Computer methods in applied mechanics and engineering* 194 (2005) 1141–1159.
- [20] T. Hughes, Multiscale phenomena: Greens’s functions, the Dirichlet-to-

- Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer methods in Applied Mechanics and engineering* 127 (1995) 387–401.
- [21] L. M. Abriola, K. Rathfelder, Mass balance errors in modeling two-phase immiscible flows: Causes and remedies, *Advances in Water Resources* 16 (1993) 223–239.
- [22] T. Hughes, G. Engel, L. Mazzei, M. Larson, The continuous Galerkin method is locally conservative, *Journal of Computational Physics* 163 (2000) 467–488.
- [23] S. Chippada, C. N. Dawson, M. L. Martinez, M. F. Wheeler, A projection method for constructing a mass conservative velocity field, *Computer Methods in Applied Mechanics and Engineering* 157 (1–2) (1998) 1–10.
- [24] G. Chavent, J. E. Roberts, A unified physical presentation of mixed, mixed-hybrid finite elements and standard finite difference approximations for the determination of velocities in waterflow problems, *Advances in Water Resources* 14 (6) (1991) 329–348.
- [25] C. Dawson, Coupling local discontinuous and continuous Galerkin methods for flow problems, *Advances in Water Resources* 28 (7) (2005) 729–744.
- [26] L. Bergamaschi, M. Putti, Mixed finite elements and Newton-type linearizations for the solution of Richards’ equation, *International Journal for Numerical Methods in Engineering* 45 (1999) 1025–1046.
- [27] H. Li, M. W. Farthing, C. T. Miller, Adaptive local discontinuous Galerkin approximation to Richards’ equation, *Advances in Water Resources* 30 (2007) 1883–1901.
- [28] C. Cordes, K. Kinzelbach, Continuous groundwater velocity field and path lines in linear, bilinear, and trilinear finite elements, *Water Resources Research* 28 (11) (1992) 2903–2911.
- [29] P. Ackerer, R. Mos, P. Siegel, G. Chavent, Reply ”Application of the mixed hybrid finite element approximation in a groundwater flow model: Luxury or necessity?” by R. Mosé, P. Siegel, P. Ackerer, and G. Chavent, *Water Resources Research* 32 (6) (1996) 1911–1913.
- [30] C. Cordes, K. Kinzelbach, Comment ”Application of the mixed hybrid finite element approximation in a groundwater flow model: Luxury or necessity?” by R. Mosé, P. Siegel, P. Ackerer, and G. Chavent, *Water Resources Research* 32 (6) (1996) 1905–1909.
- [31] L. J. Durlofsky, Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities, *Water Resources Research* 30 (4) (1994) 965–973.
- [32] S. Turek, *Efficient Solvers for Incompressible Flow Problems: An algorithmic and Computational Approach*, Vol. 6 of *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, 1999.
- [33] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, *Computer Methods in Applied Mechanics and Engineering* 156 (1998) 185–210.

- [34] L. P. Franca, S. L. Frey, T. J. R. Hughes, Stabilized finite element methods: I. application to the advective-diffusive model, *Computer Methods in Applied Mechanics and Engineering* 95 (2) (1992) 253–276.
- [35] S. Nagrath, K. E. Jansen, R. T. Lahey, Computation of incompressible bubble dynamics with a stabilized finite element level set method, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4565–4587.
- [36] T. E. Tezduyar, S. Mittal, S. E. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, *Computer Methods in Applied Mechanics and Engineering* 95 (1992) 221–242.
- [37] T. Arbogast, Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems, *SIAM Journal on Numerical Analysis* 42 (2) (2004) 576–598.
- [38] R. Codina, J. Blasco, Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales, *Computing and Visualization in Science* 4 (2002) 167–174.
- [39] T. Arbogast, Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase darcy flow, *Computational Geosciences* 6 (2002) 453–481.
- [40] R. Juanes, T. Patzek, Multiscale-stabilized solutions to one-dimensional systems of conservation laws, *Computer methods in Applied Mechanics and Engineering* 194 (2005) 2781–2805.
- [41] F. Brezzi, L. Franca, T. Hughes, A. Russo, $b = \int g$, *Computer Methods in Applied Mechanics and Engineering* 145 (1997) 329–339.
- [42] T. Arbogast, An overview of subgrid upscaling for elliptic problems in mixed form, in: Z. Chen, R. Glowinski, K. Li (Eds.), *Current Trends in Scientific Computing, Contemporary Mathematics*, AMS, 2003, pp. 21–32.
- [43] J. Aarnes, On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, *SIAM Multiscale Modeling And Simulation* 2 (3) (2004) 421–439.
- [44] Y. Chen, L. Durlofsky, M. Gerritsen, X. Wen, A coupled local-global upscaling approach for simulating highly heterogeneous formations, *Advances in Water Resources* 26 (1041–1060).
- [45] T. Y. Hou, X. H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *Journal of Computational Physics* 134 (1997) 169–189.
- [46] P. Jenny, S. H. Lee, H. A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, *Journal of Computational Physics* 187 (2003) 47–67.
- [47] C. Dawson, S. Sun, M. F. Wheeler, Compatible algorithms for coupled flow and transport, *Computer Methods in Applied Mechanics and Engineering* 193 (2004) 2565–2580.

- [48] Z. Chen, R. E. Ewing, From single-phase to compositional flow: Applicability of mixed finite elements, *Transport in Porous Media* 27 (1997) 225–242.
- [49] R. Klausen, T. Russell, Relationships among some locally conservative discretization methods which handle discontinuous coefficients, *Computational Geosciences* 8 (4) (2004) 341–377.
- [50] D. Arnold, F. Brezzi, Mixed and nonconforming finite element methods: Implementation, postprocessing, and error estimates, *Mathematical Modelling and Numerical Analysis* 19 (1) (1985) 7–32.
- [51] L. D. Marini, An inexpensive method for the evaluation of the solution of the lowest order Raviart-Thomas mixed method, *Siam Journal on Numerical Analysis* 22 (3) (1985) 493–496.
- [52] T. Arbogast, Z. Chen, On the implementation of mixed methods as nonconforming methods for second order elliptic problems, *Mathematics of Computation* 64 (211) (1995) 943–972.
- [53] Z. Chen, Equivalence between and multigrid algorithms for nonconforming and mixed methods for second order elliptic problems, *East-West Journal of Numerical Mathematics*.
- [54] C. T. Miller, G. Christakos, P. T. Imhoff, J. F. McBride, J. A. Pedit, J. A. Trangenstein, Multiphase flow and transport modeling in heterogeneous porous media: Challenges and approaches, *Advances in Water Resources* 21 (2) (1998) 77–120.
- [55] M. T. van Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Science Society of America Journal* 44 (1980) 892–898.
- [56] Y. Mualem, A new model for predicting the hydraulic conductivity of unsaturated porous media, *Water Resources Research* 12 (3) (1976) 513–522.
- [57] J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, Springer-Verlag, Berlin, Germany, 2001.
- [58] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, 1996.
- [59] G. Hauke, A simple subgrid scale stabilized method for the advection-diffusion-reaction equation, *Computer Methods in Applied Mechanics and Engineering* 191 (2002) 2925–2947.
- [60] E. Burman, A. Ern, Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation, *Computer Methods in Applied Mechanics and Engineering*.
- [61] T. J. R. Hughes, M. Mallet, A. Mizukami, A new finite element formulation for computational fluid dynamics: II. beyond SUPG, *Computer Methods in Applied Mechanics and Engineering* 54 (3) (1986) 341–355.
- [62] F. Brezzi, J. Douglas Jr., R. Durán, M. Fortin, Mixed finite elements for second order elliptic problems in three variables, *Numerische Mathematik* 51 (1987) 237–250.

- [63] Y. Bazilevs, T. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Computers & Fluids* 36 (2007) 12–26.
- [64] F. Lehmann, P. H. Ackerer, Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media, *Transport in Porous Media* 31 (3) (1998) 275–292.
- [65] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1995.
- [66] W. L. Briggs, V. E. Henson, S. F. McCormick, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [67] C. E. Kees, M. W. Farthing, E. W. Jenkins, C. T. Kelley, S. E. Howington, Multiscale models of air/water flow in porous media, in: P. J. Binning, P. K. Engesgaard, H. K. Dahle, G. F. Pinder, W. G. Gray (Eds.), *XVI International Conference on Computational Methods in Water Resources*, Copenhagen, Denmark, 2006.
- [68] X. Li, An overview of SuperLU: Algorithms, implementation, and user interface, *ACM Transactions on Mathematical Software* 31 (3) (2005) 302–325.
- [69] E. Hairer, S. P. Norsett, G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer-Verlag, Berlin, 1993.
- [70] T. Tezduyar, Y. Osawa, Finite element stabilization parameters computed from element matrices and vectors, *Computer Methods in Applied Mechanics and Engineering* 3–4 (2000) 411–430.
- [71] C. T. Miller, G. A. Williams, C. T. Kelley, M. D. Tocci, Robust solution of Richards’ equation for non-uniform porous media, *Water Resources Research* 34 (10) (1998) 2599–2610.
- [72] M. W. Farthing, C. E. Kees, C. T. Miller, Mixed finite element methods and higher-order temporal approximations, *Advances in Water Resources* 25 (1) (2002) 85–101.
- [73] K. Kang, P_1 nonconforming finite element multigrid method for radiation transport, *SIAM Journal on Scientific Computing* 25 (2) (2003) 369–384.
- [74] J. Shewchuk, Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in: M. Lin, D. Manocha (Eds.), *Applied Computational Geometry: Towards Geometric Engineering*, Vol. 1148 of *Lecture Notes in Computer Science*, Springer-Verlag, 1996, pp. 203–222, <http://www.cs.cmu.edu/~quake/triangle.html>.