Available online at www.sciencedirect.com
SCIENCE DIRECT®
Finite Elements in Analysis and Design 41 (2005) 763–777
FINITE ELEMENTS IN ANALYSIS AND DESIGN
www.elsevier.com/locate/finel

A variational multiscale finite element method for multiphase flow in porous media

Ruben Juanes

Department of Petroleum Engineering, Stanford University, 65 Green Earth Sciences Building, Stanford, CA 94305, USA

Received 9 September 2004; accepted 1 October 2004
Available online 24 January 2005

Abstract

We present a stabilized finite element method for the numerical solution of multiphase flow in porous media, based on a multiscale decomposition of pressures and fluid saturations into resolved (or grid) scales and unresolved (or subgrid) scales. The multiscale split is invoked in a variational setting, which leads to a rigorous definition of a grid scale problem and a subgrid scale problem. The subgrid problem is modeled using an algebraic approximation. This model requires the definition of a matrix of intrinsic time scales, which we design based on stability considerations.

We illustrate the performance of the method with simulations of a waterflood in a heterogeneous oil reservoir. The proposed method yields stable, highly accurate solutions on very coarse grids, which we compare with those obtained by the classical Galerkin method or the upstream finite difference method.

Although this paper is restricted to multiphase flow in porous media, the formulation is quite general and can be applied to other nonlinear systems of conservation laws, like the shallow water equations.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Finite elements; Variational multiscale; Stabilized methods; Conservation laws; Porous media; Multiphase flow; Shocks; Waterflood

E-mail address: ruben.juanes@stanford.edu.

0168-874X/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.finel.2004.10.008

1. Introduction

Numerical simulation of multiphase flow in the subsurface is an essential step in the evaluation of the performance of hydrocarbon reservoirs, and the migration of nonaqueous-phase liquids in the groundwater.

In reservoir engineering, the mathematical problem of multiphase flow has traditionally been posed in terms of a pressure equation, and a saturation equation (or a system of saturation equations if there are more than two flowing phases) [1,2]. These two equations are of very different character. The pressure equation is essentially elliptic (when the fluids and the medium are nearly incompressible), and the saturation equation is almost hyperbolic (when capillarity effects are small). Numerical models typically reflect the different characters of these equations, and employ a centered discretization of the global flux in the pressure equation, and an upstream discretization of the phase flux in the saturation equation. While this treatment of the phase flux leads to robust numerical schemes, it is well known that it introduces excessive numerical diffusion, which may distort the physical behavior inadmissibly. This is particularly true for systems in which capillarity is small.

In this paper, we present a numerical formulation for the solution of multiphase flow problems that yields stable and high-order accurate solutions in the limit of negligible capillarity. The method, which is based on an idea originally introduced in [3], entails a decomposition of the unknowns (the pressure and the fluid saturations) into resolved and unresolved scales. The multiscale decomposition recognizes that the solution contains features, such as shocks and boundary layers, which cannot be captured with a grid of practical size. The major benefit of the multiscale split is that, by incorporating the effect of the subgrid scales into the coarse-scale problem, the formulation leads naturally to a stabilized method [4–6].

The formulation presented here has the following distinctive features: (1) the multiscale split is invoked prior to any linearization of the equations; (2) the procedure does not involve reconstruction of coarse-scale and subscale solutions; and (3) we extend our previous work on one-dimensional problems to several space dimensions [7–9]. Even though we study multiphase flow problems only, the formulation is rather general, and it is applicable to other nonlinear systems of conservation laws.

An outline of the paper is as follows. In Section 2 we give the mathematical model of two-phase flow in porous media, including a derivation of the governing equations, and the weak form of the problem. In Section 3 we describe the variational multiscale method for general nonlinear systems of conservation laws of advection–diffusion type, and design the matrix of intrinsic time scales for the two-phase flow problem of interest. In Section 4 we illustrate the performance of the method with a two-dimensional problem of water injection in a simplistic heterogeneous oil reservoir. In Section 5 we make some final remarks.

2. Mathematical formulation

2.1. Governing equations of two-phase flow

In this section we outline the derivation of the equations governing immiscible flow of two species (or components) through porous media. The basic ingredients are mass conservation of each species and a constitutive relation defining the flux of each component. A detailed derivation can be found elsewhere [1,2,10].

The system involves simultaneous flow of two separate phases: the wetting phase (water) and the nonwetting phase (oil). The mass conservation equations for each component are

$$\text{Oil:} \quad \partial_t m_o + \nabla \cdot \boldsymbol{F}_o = Q_o, \tag{1}$$

$$\text{Water:} \quad \partial_t m_w + \nabla \cdot \boldsymbol{F}_w = Q_w, \tag{2}$$

where m_o (resp. m_w) is the mass of oil (resp. water) per unit volume of porous medium, $\boldsymbol{F}_o$ (resp. $\boldsymbol{F}_w$) is the oil (resp. water) mass flux, and Q_o (resp. Q_w) is the oil (resp. water) source term.

Denoting by S_w the water saturation (volume fraction of the pore space occupied by water), we express

$$m_o = \rho_o \phi (1 - S_w), \tag{3}$$

$$m_w = \rho_w \phi S_w, \tag{4}$$

where ρ_o (resp. ρ_w) is the density of the oil (resp. water), and ϕ is the porosity of the medium. The source term of each phase can be written as

$$Q_o = \rho_o \phi q_T (1 - f_w^*), \tag{5}$$

$$Q_w = \rho_w \phi q_T f_w^*, \tag{6}$$

where q_T is the total volumetric source of fluid, and f_w^* is the water fractional flow of the injected fluid. In the case of negative source ($q_T < 0$), the water fractional flow of the produced fluid is dictated by the water saturation in the reservoir ($f_w^* = f_w(S_w)$).

Mass fluxes of each component are expressed in terms of the phase velocities $\boldsymbol{v}_o$ and $\boldsymbol{v}_w$:

$$\boldsymbol{F}_o = \rho_o \phi \boldsymbol{v}_o, \tag{7}$$

$$\boldsymbol{F}_w = \rho_w \phi \boldsymbol{v}_w. \tag{8}$$

The phase velocities are usually modeled through the multiphase extension of Darcy's law [11]:

$$\boldsymbol{v}_o = -\frac{\mathbf{k}}{\phi} \frac{k_{ro}}{\mu_o} \nabla p_o, \tag{9}$$

$$\boldsymbol{v}_w = -\frac{\mathbf{k}}{\phi} \frac{k_{rw}}{\mu_o} \nabla p_w, \tag{10}$$

where k_{ro}, k_{rw} are the relative permeabilities, μ_o, μ_w are the dynamic viscosities, and p_o, p_w are the pressures of oil and water, respectively. It is assumed that the fluid pressures incorporate the hydrostatic (gravity) term. We introduce, for convenience, the phase relative mobilities:

$$\lambda_o := \frac{k_{ro}}{\mu_o}, \quad \lambda_w := \frac{k_{rw}}{\mu_w}, \tag{11}$$

and the total mobility:

$$\lambda_T := \lambda_o + \lambda_w. \tag{12}$$

We express the phase pressures as

$$p_{\mathrm{o}} \equiv p, \tag{13}$$

$$p_{\mathrm{w}} \equiv p - P_{\mathrm{c}}, \tag{14}$$

where we choose the pressure variable p to be the oil-phase pressure, and P_{c} is the capillary pressure, assumed here to be a function of saturation. Defining the total velocity $\boldsymbol{v}_{\mathrm{T}} := \boldsymbol{v}_{\mathrm{w}} + \boldsymbol{v}_{\mathrm{o}}$, the phase velocities take the form

$$\boldsymbol{v}_{\mathrm{o}} = \frac{\lambda_{\mathrm{o}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{o}}} \boldsymbol{v}_{\mathrm{T}} - \frac{\lambda_{\mathrm{w}} \lambda_{\mathrm{o}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{o}}} \frac{\mathbf{k}}{\phi} \nabla P_{\mathrm{c}}, \tag{15}$$

$$\boldsymbol{v}_{\mathrm{w}} = \frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{o}}} \boldsymbol{v}_{\mathrm{T}} + \frac{\lambda_{\mathrm{w}} \lambda_{\mathrm{o}}}{\lambda_{\mathrm{w}} + \lambda_{\mathrm{o}}} \frac{\mathbf{k}}{\phi} \nabla P_{\mathrm{c}}. \tag{16}$$

We also define the oil and water fractional flow functions

$$f_{\mathrm{o}} = \frac{\lambda_{\mathrm{o}}}{\lambda_{\mathrm{T}}}, \quad f_{\mathrm{w}} = \frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{T}}}. \tag{17}$$

Assuming that the fluids and the medium are incompressible, summing Eqs. (1) and (2) yields the continuity (or "pressure") equation

$$\nabla \cdot \boldsymbol{v}_{\mathrm{T}} = q_{\mathrm{T}}, \tag{18}$$

where the total velocity is obtained by adding the phase velocities (9)–(10), and using the phase pressures (13)–(14):

$$\boldsymbol{v}_{\mathrm{T}} = -\lambda_{\mathrm{T}} \frac{\mathbf{k}}{\phi} \nabla p - \lambda_{\mathrm{w}} \frac{\mathbf{k}}{\phi} \left(-\frac{\mathrm{d}P_{\mathrm{c}}}{\mathrm{d}S_{\mathrm{w}}} \right) \nabla S_{\mathrm{w}}. \tag{19}$$

Substituting Eqs. (4), (8) and (16) into (2), we get the "saturation" equation

$$\partial_t S_{\mathrm{w}} + \nabla \cdot \left[\boldsymbol{v}_{\mathrm{T}} f_{\mathrm{w}} - \lambda_{\mathrm{w}} f_{\mathrm{o}} \frac{\mathbf{k}}{\phi} \left(-\frac{\mathrm{d}P_{\mathrm{c}}}{\mathrm{d}S_{\mathrm{w}}} \right) \nabla S_{\mathrm{w}} \right] = q_{\mathrm{T}} f_{\mathrm{w}}^{*}. \tag{20}$$

We make the following observations:

(1) The "pressure" equation is elliptic, and has to be solved for the pressure p and the total velocity $\boldsymbol{v}_{\mathrm{T}}$.
(2) The "saturation" equation is a conservation law of the advection–diffusion type, where the "diffusion" in the saturation equation is caused by capillarity, not by physical diffusion.
(3) Both equations are nonlinear due to the nonlinearity of relative permeabilities and capillary pressure.
(4) The pressure and saturation equations are coupled through the mobility terms and the capillary pressure terms.

Eqs. (18) and (20) can be formulated as a system of equations in *parabolic form*:

$$\begin{bmatrix} 0 \\ \partial_t S_{\mathrm{w}} \end{bmatrix} + \nabla \cdot \left(-\frac{\mathbf{k}}{\phi} \begin{bmatrix} \lambda_{\mathrm{T}} & \lambda_{\mathrm{w}}(-P_{\mathrm{c}}') \\ \lambda_{\mathrm{w}} & \lambda_{\mathrm{w}}(-P_{\mathrm{c}}') \end{bmatrix} \begin{bmatrix} \nabla p \\ \nabla S_{\mathrm{w}} \end{bmatrix} \right) = \begin{bmatrix} q_{\mathrm{T}} \\ q_{\mathrm{T}} f_{\mathrm{w}}^{*} \end{bmatrix}, \tag{21}$$

and in *hyperbolic form*:

$$\begin{bmatrix} 0 \\ \partial_t S_{\mathrm{w}} \end{bmatrix} + \nabla \cdot \left(\begin{bmatrix} \mathbf{0} \\ \boldsymbol{v}_{\mathrm{T}} f_{\mathrm{w}} \end{bmatrix} - \frac{\mathbf{k}}{\phi} \begin{bmatrix} \lambda_{\mathrm{T}} & \lambda_{\mathrm{w}}(-P_{\mathrm{c}}') \\ 0 & \lambda_{\mathrm{w}}(1-f_{\mathrm{w}})(-P_{\mathrm{c}}') \end{bmatrix} \begin{bmatrix} \nabla p \\ \nabla S_{\mathrm{w}} \end{bmatrix} \right) = \begin{bmatrix} q_{\mathrm{T}} \\ q_{\mathrm{T}} f_{\mathrm{w}}^{*} \end{bmatrix}. \tag{22}$$

In either case, the system of equations may be expressed in the generic form of a nonlinear system of conservation laws of advection–diffusion type:

$$\hat{\partial}_t \boldsymbol{u} + \nabla \cdot (\boldsymbol{f}(\boldsymbol{u}) - \mathbf{D}(\boldsymbol{u}) \nabla \boldsymbol{u}) = \boldsymbol{q}, \tag{23}$$

where $\boldsymbol{u} = (p, S_{\mathrm{w}})^{\mathrm{T}}$ is the vector of unknowns, $\boldsymbol{f}$ is the hyperbolic flux, $\mathbf{D}$ is the diffusion tensor, $\boldsymbol{q}$ is the source term, and $\hat{\partial}_t$ implies that the first component of the accumulation term is identically equal to zero. In what follows, we shall abuse notation and drop the hat from the time derivative, and still understand that the first component of this vector is zero.

For clarity, Eq. (23) can be written in indicial notation

$$\frac{\partial u^i}{\partial t} + \sum_{k=1}^{d} \frac{\partial}{\partial x_k} \left(f_k^i - \sum_{j=1}^{2} \sum_{l=1}^{d} D_{kl}^{ij} \frac{\partial u^j}{\partial x_l} \right) = q^i, \quad i = 1, 2, \tag{24}$$

where superscripts i, j refer to components (pressure and water saturation), and subscripts k, l refer to space dimensions ($d = 2$ or 3).

2.2. Initial and boundary value problem

The mathematical problem is defined by the system of conservation laws:

$$\partial_t \boldsymbol{u} + \nabla \cdot (\boldsymbol{f}(\boldsymbol{u}) - \mathbf{D}(\boldsymbol{u}) \nabla \boldsymbol{u}) = \mathbf{0}, \quad \boldsymbol{x} \in \Omega, \quad t \in (0, T]. \tag{25}$$

Let $\partial\Omega$ be the boundary of the domain. We consider, for expositional simplicity, the following boundary conditions:

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on } \Gamma_u, \tag{26}$$

$$(\boldsymbol{f} - \mathbf{D}\nabla \boldsymbol{u}) \cdot \boldsymbol{n} = \bar{F} \quad \text{on } \Gamma_n \equiv \partial\Omega \backslash \Gamma_u, \tag{27}$$

where $\boldsymbol{n}$ is the outward unit normal to the boundary. The method can accommodate much more complex boundary conditions and, in particular, conditions of different types for each individual component of the solution. The initial conditions are

$$\boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{u}_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \bar{\Omega}. \tag{28}$$

2.3. Weak form of the problem

The weak form of problem (25)–(28) is obtained in the standard way, by multiplying both sides of the partial differential equation by a suitable test function $\boldsymbol{v}$, integrating over the domain Ω, and integrating the flux term by parts.

We introduce the following functional spaces:

$$\mathscr{V} := \{\boldsymbol{v} \in W : \boldsymbol{v} = \bar{\boldsymbol{u}} \text{ on } \Gamma_u\},$$
$$\mathscr{V}_0 := \{\boldsymbol{v} \in W : \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_u\},$$

where the appropriate Sobolev space W depends on the particular form of the diffusion tensor. The weak form of problem (25)–(28) consists in finding $u \in \mathscr{V}$ for each fixed $t \in (0, T]$, such that

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) + a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{u}) = l(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathscr{V}_0,$$
$$\boldsymbol{u}(x, t = 0) = \boldsymbol{u}_0(x), \tag{29}$$

where

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \partial_t \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}\Omega, \tag{30}$$

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{w}) = -\int_\Omega \boldsymbol{f}(\boldsymbol{w}) \cdot \nabla \boldsymbol{v} \, \mathrm{d}\Omega + \int_\Omega \mathbf{D}(\boldsymbol{w}) \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} \, \mathrm{d}\Omega, \tag{31}$$

$$l(\boldsymbol{v}) = \int_\Omega \boldsymbol{q} \cdot \boldsymbol{v} \, \mathrm{d}\Omega - \int_{\Gamma_n} \bar{\boldsymbol{F}} \cdot \boldsymbol{v} \, \mathrm{d}\Gamma. \tag{32}$$

When expressed in indices, Eqs. (30)–(32) above take the form

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \sum_{i=1}^{2} \partial_t u^i v^i \, \mathrm{d}\Omega, \tag{33}$$

$$a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{w}) = -\int_\Omega \sum_{i=1}^{2} \sum_{k=1}^{d} f_k^i(\boldsymbol{w}) \frac{\partial v^i}{\partial x_k} \, \mathrm{d}\Omega + \int_\Omega \sum_{i=1}^{2} \sum_{k=1}^{d} \left(\sum_{j=1}^{2} \sum_{l=1}^{d} D_{kl}^{ij}(\boldsymbol{w}) \frac{\partial u^j}{\partial x_l} \right) \frac{\partial v^i}{\partial x_k} \, \mathrm{d}\Omega, \tag{34}$$

$$l(\boldsymbol{v}) = \int_\Omega \sum_{i=1}^{2} q^i v^i \, \mathrm{d}\Omega - \int_{\Gamma_n} \sum_{i=1}^{2} \bar{F}^i v^i \, \mathrm{d}\Gamma. \tag{35}$$

3. The variational multiscale method

In order to solve the problem numerically, we must restrict the infinite-dimensional functional spaces of trial and test functions to finite sets. In particular, we consider spaces whose functions are linear combinations of the usual finite element basis functions, defined on a partition of the domain into elements $\Omega \approx \bigcup \Omega_e$. In the *classical Galerkin method*, the same basis functions are used to define the trial functions $\boldsymbol{u}_h$ and test functions $\boldsymbol{v}_h$. The subscript h refers to the resolution provided by the underlying grid, that is, the characteristic element size.

Obviously, the numerical solution cannot capture features at a scale smaller than the gridblock size. Such features may be due to subgrid heterogeneity or to the presence of sharp fronts. As it turns out, inaccuracies at the subgrid level may resonate, and produce a numerical solution that is globally polluted with errors. This is the case in nearly hyperbolic problems, even if the medium is homogeneous. The key

idea of the *variational multiscale method* [3] is to recognize the limited resolution of the grid, and invoke a *multiscale split* of the solution $\boldsymbol{u}$ and the test function $\boldsymbol{v}$:

$$\boldsymbol{u} = \boldsymbol{u}_h + \tilde{\boldsymbol{u}}, \quad \boldsymbol{v} = \boldsymbol{v}_h + \tilde{\boldsymbol{v}}, \tag{36}$$

where $\boldsymbol{u}_h$ is the resolved (grid) scale and $\tilde{\boldsymbol{u}}$ is the unresolved (subgrid) scale. This decomposition is unique if the original functional space $\mathscr{V}$ can be obtained as the direct sum of two spaces:

$$\mathscr{V} = \mathscr{V}_h \oplus \tilde{\mathscr{V}}, \tag{37}$$

where $\mathscr{V}_h$ is the space of *resolved scales* and $\tilde{\mathscr{V}}$ is the space of *subgrid scales*. From a practical point of view, the space $\tilde{\mathscr{V}}$ is generally unknown, and needs to be approximated.

In the context of nonlinear problems, we express the solution at a given iteration step (k) as

$$\boldsymbol{u}^{(k)} = \boldsymbol{u}^{(k-1)} + \delta\boldsymbol{u}^{(k-1)}, \tag{38}$$

where the first term on the right-hand side is an approximate solution at the previous iteration level, and the second term is a correction. In principle, both terms are subject to the multiscale decomposition (36). However, doing so requires that the approximate solution $\boldsymbol{u}^{(k-1)}$ is reconstructed after every iteration. To avoid this reconstruction step, we make the additional approximation:

$$\boldsymbol{u}^{(k-1)} \approx \boldsymbol{u}_h^{(k-1)}, \tag{39}$$

so that the multiscale split takes the form

$$\boldsymbol{u}^{(k)} \approx \boldsymbol{u}_h^{(k)} + \delta\tilde{\boldsymbol{u}}^{(k-1)}. \tag{40}$$

In what follows we shall drop superscripts referring to the iteration level, and simply write

$$\boldsymbol{u} \approx \boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}}. \tag{41}$$

The term $\boldsymbol{u}_h$ should be understood as an approximate solution about which the equations are linearized, and the term $\delta\tilde{\boldsymbol{u}}$ plays the role of a subgrid scale perturbation. The working assumption (39) makes our formulation different from that in [12–14], where the subgrid scales are tracked, and the multiscale variable is reconstructed after every step of the iterative process. It is likely, however, that the proposed linearization may impose limits in the applicability of the formulation to highly nonlinear problems with a very strong multiscale character.

We substitute the multiscale decomposition (41) in the weak form of problem (29). Since the weak form is linear with respect to the test function $\boldsymbol{v}$, we can split the original problem into two, a grid scale problem:

$$(\partial_t(\boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}}), \boldsymbol{v}_h) + a(\boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}},\ \boldsymbol{v}_h; \boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}}) = l(\boldsymbol{v}_h) \quad \forall \boldsymbol{v}_h \in \mathscr{V}_{h,0}, \tag{42}$$

and a subscale problem:

$$(\partial_t(\boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}}), \tilde{\boldsymbol{v}}) + a(\boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}},\ \tilde{\boldsymbol{v}}; \boldsymbol{u}_h + \delta\tilde{\boldsymbol{u}}) = l(\tilde{\boldsymbol{v}}) \quad \forall \tilde{\boldsymbol{v}} \in \tilde{\mathscr{V}}. \tag{43}$$

The grid scale problem is finite-dimensional (the dimensionality given, roughly, by the number of nodes in the grid), whereas the subscale problem is infinite-dimensional.

3.1. Subgrid scale problem

The subscale problem cannot, in general, be solved exactly. We make the following manipulations to the subscale equation (43) (see [15,9] for details): (1) write the flux term as a sum of element integrals, and integrate by parts on each element; (2) assume continuity of the flux across interelement boundaries; (3) approximate the total flux $\boldsymbol{f}(\boldsymbol{u}) - \mathbf{D}(\boldsymbol{u})\nabla\boldsymbol{u}$ by its first-order Taylor expansion about the coarse-scale solution $\boldsymbol{u}_h$; and (4) assume quasi-static subscales, $\partial_t(\delta\tilde{\boldsymbol{u}}) \approx 0$. After these approximations, the subscale problem can be written as the following projection problem:

$$\sum_e \int_{\Omega^e} \mathscr{L}_{u_h}\delta\tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{v}}\,\mathrm{d}\Omega = \sum_e \int_{\Omega^e} \mathscr{R}(\boldsymbol{u}_h) \cdot \tilde{\boldsymbol{v}}\,\mathrm{d}\Omega. \tag{44}$$

In Eq. (44), we have introduced the grid scale residual:

$$\mathscr{R}(\boldsymbol{u}_h) := \boldsymbol{q} - \partial_t\boldsymbol{u}_h - \nabla\cdot(\boldsymbol{f}(\boldsymbol{u}_h) - \mathbf{D}(\boldsymbol{u}_h)\nabla\boldsymbol{u}_h), \tag{45}$$

and the following linearized advection–diffusion operator in conservation form:

$$\mathscr{L}_{u_h}\boldsymbol{v} := \nabla\cdot(\mathbf{A}(\boldsymbol{u}_h)\boldsymbol{v} - \mathbf{D}(\boldsymbol{u}_h)\nabla\boldsymbol{v}), \tag{46}$$

where $\mathbf{A}(\boldsymbol{u}_h)$ is an "advection" operator, defined as

$$\mathbf{A}(\boldsymbol{u}_h) := \boldsymbol{f}'(\boldsymbol{u}_h) - \mathbf{D}'(\boldsymbol{u}_h)\nabla\boldsymbol{u}_h. \tag{47}$$

For definiteness, it is useful to write down the equations above using indicial notation. The i-component of the grid scale residual is

$$\mathscr{R}^i(\boldsymbol{u}_h) = q^i - \partial_t u_h^i - \sum_{k=1}^{d}\frac{\partial}{\partial x_k}\left(f_k^i(\boldsymbol{u}_h) - \sum_{j=1}^{2}\sum_{l=1}^{d} D_{kl}^{ij}(\boldsymbol{u}_h)\frac{\partial u_h^j}{\partial x_l}\right), \tag{48}$$

The i-component of the linearized advection–diffusion operator reads

$$\mathscr{L}_{u_h}^i\boldsymbol{v} = \sum_{k=1}^{d}\frac{\partial}{\partial x_k}\left(\sum_{j=1}^{2} A_k^{ij}(\boldsymbol{u}_h)v^j - \sum_{j=1}^{2}\sum_{l=1}^{d} D_{kl}^{ij}(\boldsymbol{u}_h)\frac{\partial v^j}{\partial x_l}\right), \tag{49}$$

where the components of the advection operator are

$$A_k^{ij}(\boldsymbol{u}_h) = \frac{\partial f_k^i(\boldsymbol{u}_h)}{\partial u_h^j} - \sum_{n=1}^{2}\sum_{l=1}^{d}\frac{\partial D_{kl}^{in}(\boldsymbol{u}_h)}{\partial u_h^j}\frac{\partial u_h^n}{\partial x_l}. \tag{50}$$

The projection problem (44) is *modeled*, rather than solved, by an algebraic subgrid scale (ASGS) approximation [3,16]

$$\delta\tilde{\boldsymbol{u}} \approx \boldsymbol{\tau}_{u_h}\mathscr{R}(\boldsymbol{u}_h), \tag{51}$$

where $\boldsymbol{\tau}_{u_h}$ is the matrix of stabilizing coefficients, or matrix of intrinsic time scales. Its design is one of the most difficult issues in the development of a stabilized method. In this paper, we devise a τ-model tailored to the problem of multiphase flow.

3.2. Grid scale problem

After several manipulations and approximations, similar to those of the subscale problem (see [15,9] for details), the grid scale equation reads

$$(\partial_t \boldsymbol{u}_h, \boldsymbol{v}_h) + a(\boldsymbol{u}_h, \boldsymbol{v}_h; \boldsymbol{u}_h) + \sum_e \int_{\Omega^e} \mathscr{L}^*_{u_h} \boldsymbol{v}_h \cdot \delta\tilde{\boldsymbol{u}} \, \mathrm{d}\Omega + \sum_e \int_{\Gamma^e} \boldsymbol{b}^*_{u_h} \boldsymbol{v}_h \cdot \delta\tilde{\boldsymbol{u}} \, \mathrm{d}\Gamma = l(\boldsymbol{v}_h)$$
$$\forall \boldsymbol{v}_h \in \mathscr{V}_{h,0}. \tag{52}$$

where $\mathscr{L}^*_{u_h}$ is the adjoint of a linearized advection–diffusion operator:

$$\mathscr{L}^*_{u_h} \boldsymbol{v} := -\mathbf{A}^{\mathrm{T}}(\boldsymbol{u}_h) \cdot \nabla \boldsymbol{v} - \nabla \cdot (\mathbf{D}^{\mathrm{T}}(\boldsymbol{u}_h) \nabla \boldsymbol{v}), \tag{53}$$

and $\boldsymbol{b}^*_{u_h}$ is the associated boundary operator:

$$\boldsymbol{b}^*_{u_h} \boldsymbol{v} := (\mathbf{D}^{\mathrm{T}}(\boldsymbol{u}_h) \nabla \boldsymbol{v}) \cdot \boldsymbol{n}. \tag{54}$$

The last two terms in Eq. (52) are the contributions of the subgrid scale to the coarse-scale equation. These additional terms, which arise naturally in the variational multiscale method, stabilize the numerical approximation of nearly hyperbolic systems. They consist in a volume integral and a boundary integral, which are evaluated element by element. The boundary contribution is neglected in the calculations of Section 4.

The grid scale equation (52) and the subgrid scale equation (43) are coupled through the value of the subscales. For the algebraic subgrid scale model employed here, Eq. (51) is directly substituted in Eq. (52). If the presence of the subscales is neglected ($\delta\tilde{\boldsymbol{u}} \equiv \boldsymbol{0}$), the formulation reduces to the classical Galerkin method. Our formulation is different from other multiscale approximations [12,14] in that: (1) the nonlinearity of the equations is retained at the time of invoking the multiscale split; and (2) the approximate solution is never reconstructed from pointwise values of the coarse- and subgrid-scale components.

3.3. Matrix of intrinsic time scales

The description of the variational multiscale method proposed is complete up to the definition of the matrix $\boldsymbol{\tau}_{u_h}$ of intrinsic time scales. This matrix needs to be defined for each individual problem, and its design is the crux of any stabilized method. Here, we concentrate on the problem of two-phase flow in porous media. In principle, the matrix of stabilizing parameters is a full 2×2 matrix,

$$\boldsymbol{\tau}_{u_h} = \begin{bmatrix} \tau_{\mathrm{pp}} & \tau_{\mathrm{pw}} \\ \tau_{\mathrm{wp}} & \tau_{\mathrm{ww}} \end{bmatrix}, \tag{55}$$

where the coefficients depend on the system parameters, the element size, and the grid scale solution. A simple stability analysis for the system of equations (21) or (22) in the limit of negligible capillarity, shows that the classical Galerkin method provides control over the pressure, but that the water saturation is out of control. Therefore, we take the $\boldsymbol{\tau}$-matrix as

$$\boldsymbol{\tau}_{u_h} = \begin{bmatrix} 0 & 0 \\ 0 & \tau_{\mathrm{ww}} \end{bmatrix}. \tag{56}$$

From a physical standpoint, the $\boldsymbol{\tau}$-matrix above implies that we neglect the subgrid scale component of the pressure field, that is, $\tilde{p} \equiv 0$. This simplification is sensible because the pressure is governed by an elliptic equation, for which the classical Galerkin method is stable. The scalar stabilizing coefficient τ_{ww} corresponding to the water saturation field can be devised using different methods, originally developed for linear advection–diffusion systems [16–18]. Extension to the nonlinear system of saturation equations is straightforward once the linearized advection–diffusion operator (46) has been defined.

We define the "advective velocity" of the water saturation $\boldsymbol{a}(\boldsymbol{u}_h)$, with components

$$a_k = A_k^{22}, \quad k = 1, \ldots, d \tag{57}$$

and the associated "diffusion tensor" $\mathbf{d}(\boldsymbol{u}_h)$, with components

$$d_{kl} = D_{kl}^{22}, \quad k, l = 1, \ldots, d. \tag{58}$$

We also introduce the following quantities:

$$v = \left(\sum_{k=1}^{d} a_k a_k \right)^{1/2}, \tag{59}$$

$$\varepsilon = \left(\sum_{k=1}^{d} \sum_{l=1}^{d} d_{kl} d_{kl} \right)^{1/2}. \tag{60}$$

A model for the stabilizing coefficient, based on the original formulation in [19,17] is given by

$$\tau_{\mathrm{ww}} = \frac{1}{2} h \frac{\xi(\alpha)}{v}, \tag{61}$$

where α is a measure of the element Peclet number:

$$\alpha = \frac{1}{2} \frac{vh}{\varepsilon}, \tag{62}$$

and ξ is the diffusion corrector factor:

$$\xi(\alpha) = \coth(\alpha) - \frac{1}{\alpha}. \tag{63}$$

In the equations above, h is a characteristic element size, typically along the direction of the advective velocity. Eq. (61) is motivated by being consistent with the optimal definition of the stabilizing coefficient in the one-dimensional case, which leads to a nodally exact solution for the steady-state, linear advection–diffusion problem.

Different formulations of the intrinsic time scale coefficient exist. For example, based on an analysis of the discrete maximum principle in the stationary, one-dimensional case, the following formula was proposed in [16]:

$$\tau_{\mathrm{ww}} = \left(\frac{4}{h^2} \varepsilon + \frac{2}{h} v \right)^{-1}. \tag{64}$$

Currently, we tend to favor the original formulation presented in [17]. In any case, since the advection–diffusion operator depends on the coarse-scale solution, the matrix of stabilizing coefficients needs to be evaluated at each integration point, at every iteration.

4. Representative numerical simulations

In this section, we present an application of the variational multiscale method to multiphase flow in porous media. Numerical solutions are obtained using three different methods: the classical Galerkin finite element method, an upstream finite difference method, and the variational multiscale method with an algebraic subgrid approximation of the subscales. The objective of this section is twofold: (1) demonstrate the enhanced stability of the multiscale method, compared with the classical Galerkin method; and (2) show that the proposed method is much less diffusive than the traditional finite difference upstream discretization.

We confine our attention to a two-dimensional, two-phase waterflood problem, where water is injected in an oil reservoir. The geometry of the domain is shown in Fig. 1. It is a quarter of a five-spot pattern, with the injector located at the lower-left corner, and the producer at the upper-right corner. The medium has a high-conductivity patch, where the permeability is two orders of magnitude greater than in the rest of the domain.

We express the physical problem in *dimensionless* quantities by taking the following values of the parameters:

$$L = 1, \quad \frac{\mathbf{k}}{\phi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mu_w = 1, \quad \mu_o = 2. \tag{65}$$

We use the following relative permeability functions:

$$k_{rw} = \left(\frac{S_w - S_{wc}}{1 - S_{wc}} \right)^2, \quad S_{wc} = 0.15, \tag{66}$$

$$k_{ro} = \beta_o \frac{S_o - S_{om}}{1 - S_{om}} + (1 - \beta_o) \left(\frac{S_o - S_{om}}{1 - S_{om}} \right)^2, \quad \beta_o = 0.1, \quad S_{om} = 0.20. \tag{67}$$

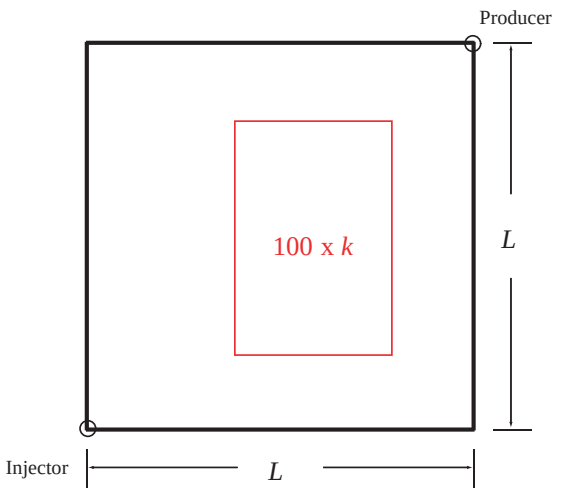

Fig. 1. Geometry of the waterflood problem.

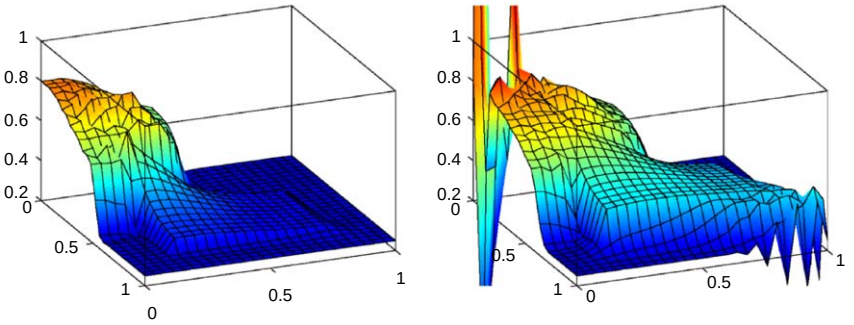

Fig. 2. Water saturation at $t = 2$ and 5, obtained with the Galerkin method.

The problem is intended to be solved in the limit of zero capillarity, and we do so when we use the upstream finite difference method. The Galerkin method, however, is not capable of dealing with the purely hyperbolic problem. Therefore, we introduce a very crude capillary pressure model, by assuming that $\lambda_w P'_{cw} = \varepsilon_w = \text{const}$. Although this model is not based on physical considerations, it is valid from a practical viewpoint because the level of capillary diffusion is sufficiently small.

The boundaries of the domain are taken as no-flow boundaries. We consider that both wells operate under pressure control. The dimensionless pressures at the wells are: $p_{inje} = 1$ at the injector, and $p_{prod} = 0$ at the producer. Only water is injected at the injection well. Therefore, the water saturation quickly reaches a constant value $S_{w,inje} = 1 - S_{om} = 0.8$. The natural boundary condition for the production well is to impose that the reservoir fluids (oil and water) are produced according to their mobility. This boundary condition was used in the finite difference calculations (performed with the commercial simulator Eclipse 100 [20]). In the Galerkin and the multiscale simulations, however, we used a Dirichlet boundary condition at the production well: $S_{w,prod} = S_{w,init}$. Such condition makes the problem more challenging numerically because a sharp boundary layer forms after breakthrough of the injected water (see below). The water saturation is initialized to a constant value, $S_{w,init} = 0.25$. The pressure variable does not require initial conditions because it is governed by an elliptic equation.

The domain is discretized into a fairly coarse grid of 25×25 elements. We use a Crank–Nicolson time-stepping technique, with a constant time step $\delta t = 0.04$. The capillary diffusion level is set to a small value $\varepsilon_w = 0.001$.

Water saturation field. The numerical solution of water saturation obtained with each of the three methods is presented below. In Fig. 2 we show the water saturation computed by the classical Galerkin method at simulation times $t = 2$ (before breakthrough) and $t = 5$ (after breakthrough). The solution is polluted with global spurious oscillations, which become especially severe after breakthrough. This behavior was expected, in view of the lack of stability of the classical Galerkin method for nearly hyperbolic problems.

The results from the upstream finite difference method (obtained using the reservoir simulator Eclipse 100) are shown in Fig. 3. Even though the problem has zero capillarity and should display sharp moving fronts, the numerical solution is extremely diffusive.

In Fig. 4 we plot the water saturation obtained with the variational multiscale formulation proposed in this paper. By incorporating the effect of the subscales in the coarse-scale equation, the method removes

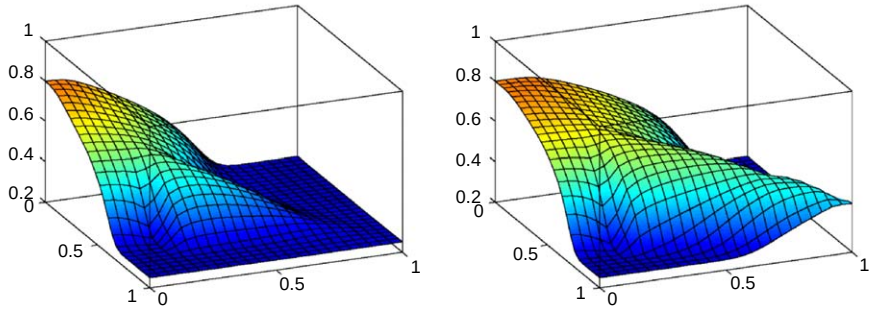

Fig. 3. Water saturation at $t = 2$ and 5, obtained with the upstream FD method.

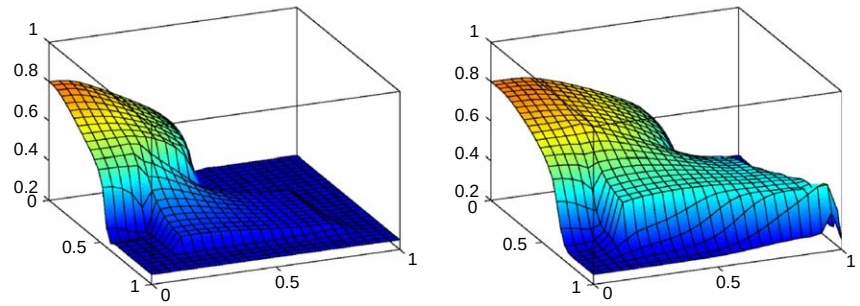

Fig. 4. Water saturation at $t = 2$ and 5, obtained with the multiscale method.

the oscillatory behavior of the classical Galerkin method, and produces a stable solution. The accuracy of the solution is also remarkable. Despite the fact that a very coarse mesh was used, the method is able to reproduce all the sharp features of the solution: the shock of invading water, the boundary layer that forms at the interface between the high- and low-permeability regions, and the boundary layer near the production well after breakthrough. It is important to emphasize at this point that the computational cost of the multiscale method is only marginally higher than that of the classical Galerkin method. The difference reduces to the computation of few additional integrals, evaluated element by element. In fact, this additional computational cost is sometimes offset by the improved convergence behavior of the nonlinear iteration at every time step.

Pressure field. By using the τ-matrix proposed in Eq. (56), we are effectively neglecting the pressure subscales. However, the pressure solution of the Galerkin method and the multiscale method are not the same. By virtue of the strong coupling between the pressure and saturation equations, the saturation subscales impact the pressure solution. This effect is illustrated in Figs. 5 and 6, where we plot the pressure solution obtained with the Galerkin method and the multiscale method, respectively, at times $t = 2$ and 5.

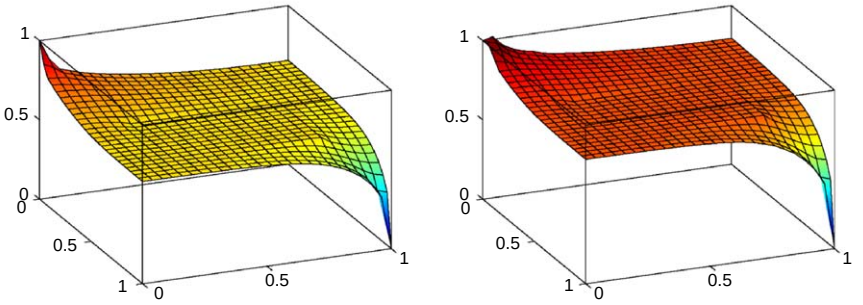

Fig. 5. Pressure at $t = 2$ and 5, obtained with the classical Galerkin method.

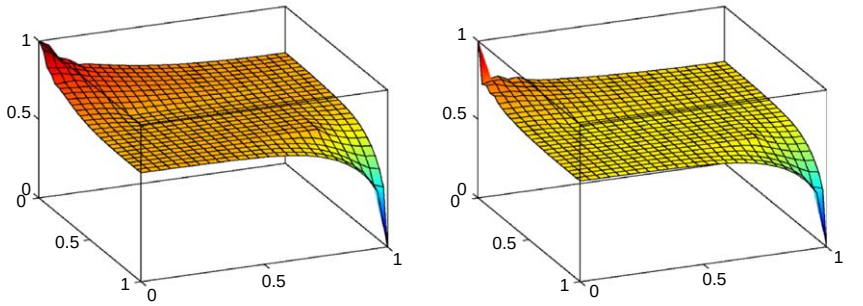

Fig. 6. Pressure at $t = 2$ and 5, obtained with the multiscale method.

The Galerkin solution and the multiscale solution are very different, especially after breakthrough. Refinement studies confirm that the multiscale solution is highly accurate.

5. Conclusions

We have presented a variational multiscale method for the numerical simulation of multiphase flow in porous media. The multiscale formulation, which is based on the original framework of [3], leads naturally to a stabilized finite element method. The formulation was derived for general nonlinear systems of conservation laws of advection–diffusion type, extending our previous work on one-dimensional systems [7,8]. The design of the matrix of intrinsic time scales, however, is case-dependent. The proposed structure of this matrix (for the two-phase flow problem of interest) stems from an examination of the lack of stability of the Galerkin method.

We demonstrate the excellent performance of the method with an illustrative example involving a waterflood of a heterogeneous oil reservoir. The variational multiscale method yields a stable and highly accurate numerical solution of both the pressure and the water saturation on a very coarse grid. The multiscale solution does not present spurious oscillations, and captures the sharp features of the physical solution. Although not reported here, the proposed multiscale method yields equally successful results when applied to other problems of practical interest, such as the shallow water equations.

References

[1] K. Aziz, A. Settari, Petroleum Reservoir Simulation, Elsevier, London, 1979.
[2] G. Chavent, J. Jaffré, Mathematical Models and Finite Elements for Reservoir Simulation, Studies in Mathematics and its Applications, vol. 17, Elsevier, North-Holland, 1986.
[3] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to- Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Eng. 127 (1995) 387–401.
[4] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1998) 3–24.
[5] F. Brezzi, L.P. Franca, T.J.R. Hughes, A. Russo, $b = \int g$, Comput. Methods Appl. Mech. Eng. 145 (1997) 329–339.
[6] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, Comput. Methods Appl. Mech. Eng. 156 (1998) 185–210.
[7] R. Juanes, T.W. Patzek, Multiscale-stabilized finite element methods for miscible and immiscible flow in porous media, J. Hydraul. Res. 42 (Sp.Iss.) (2004) 131–140.
[8] R. Juanes, T.W. Patzek, Multiscale numerical modeling of three-phase flow, in: SPE Annual Technical Conference and Exhibition, Denver, CO, 2003, (SPE 84369) Soc. Petro. Eng. J., (submitted for publication).
[9] R. Juanes, T. W. Patzek, Multiscale-stabilized solutions to one-dimensional systems of conservation laws, Comput. Methods Appl. Mech. Eng. (in press).
[10] D.W. Peaceman, Fundamentals of Numerical Reservoir Simulation, Developments in Petroleum Science, vol. 6, Elsevier, Amsterdam, 1977.
[11] J. Bear, Dynamics of Fluids in Porous Media, Environmental Science Series, Elsevier, New York, 1972 (reprinted with corrections, Dover, New York, 1988).
[12] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Eng. 191 (39–40) (2002) 4295–4321.
[13] K. Garikipati, T.J.R. Hughes, A study of strain localization in a multiple scale framework—the one-dimensional problem, Comput. Methods Appl. Mech. Eng. 159 (1998) 193–222.
[14] K. Garikipati, T.J.R. Hughes, A variational multiscale approach to strain localization—formulation for multidimensional problems, Comput. Methods Appl. Mech. Eng. 188 (2000) 39–60.
[15] R. Juanes, Displacement theory and multiscale numerical modeling of three-phase flow in porous media, Ph.D. Dissertation, University of California at Berkeley, March 2003.
[16] R. Codina, On stabilized finite element methods for linear systems of convection-diffusion-reaction equations, Comput. Methods Appl. Mech. Eng. 188 (2000) 61–82.
[17] T.J.R. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective–diffusive systems, Comput. Methods Appl. Mech. Eng. 58 (1986) 305–328.
[18] F. Shakib, T.J.R. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 89 (1991) 141–219.
[19] T.J.R. Hughes, M. Mallet, A. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG, Comput. Methods Appl. Mech. Eng. 54 (1986) 341–355.
[20] Schlumberger, Eclipse Technical Description, v. 2003A.