Multiscale Model. Simul.
Vol. 12, No. 4, pp. 1401–1423

A GLOBAL JACOBIAN METHOD FOR MORTAR DISCRETIZATIONS OF A FULLY IMPLICIT TWO-PHASE FLOW MODEL*

BENJAMIN GANIS†‡, KUNDAN KUMAR†, GERGINA PENCHEVA†, MARY F. WHEELER†, AND IVAN YOTOV§

Abstract. We consider a fully implicit formulation for two-phase flow in a porous medium with capillarity, gravity, and compressibility in three dimensions. The method is implicit in time and uses the multiscale mortar mixed finite element method for a spatial discretization in a nonoverlapping domain decomposition context. The interface conditions between subdomains are enforced in terms of Lagrange multiplier variables defined on a mortar space. The novel approach in this work is to linearize the coupled system of subdomain and mortar variables simultaneously to form a global Jacobian. This algorithm is shown to be more efficient and robust compared to previous algorithms that relied on two separate nested linearizations of subdomain and interface variables. We also examine various upwinding methods for accurate integration of phase mobility terms near subdomain interfaces. Numerical tests illustrate the computational benefits of this scheme.

Key words. global linearization, two-phase flow, porous media flow, nonoverlapping domain decomposition, multiscale method, mortar finite element, mixed finite element

AMS subject classifications. 65M55, 65M60, 76S05

DOI. 10.1137/140952922

1. Introduction.

This work develops algorithmic improvements for multiscale modeling of two-phase flow in a porous medium, which is an important area for energy and environmental applications. These problems can be very challenging to model with strong heterogeneities in the data, degeneracies occurring when residual saturations are reached, and in the evolution of sharp fronts. Multiscale methods remain a field of intensive research due to the range of spatial and temporal scales involved, geologic complexity, and the need to perform local parallel computation on large domains. One of the distinguishing features in our formulation is the ability to include the physical effects of capillarity, gravity, and compressibility, which also play an important role in two-phase computations. The key idea in algorithm development is to linearize the coupled system in all variables to form a global Jacobian, providing a useful extension of a previous work [14] to the two-phase fully implicit setting.

The class of methods under consideration are called multiscale mortar methods [4], which are a type of nonoverlapping domain decomposition method [32, 43, 34]. Following a "divide and conquer" strategy, the global problem is decomposed into

*Received by the editors January 16, 2014; accepted for publication (in revised form) June 30, 2014; published electronically October 9, 2014. An earlier version of this paper appears as ICES Report 14-01, available online at http://www.ices.utexas.edu/media/reports/2014/1401.pdf. This material is based upon work supported by NSF CDI grant DMS 0835745, and the Center for Frontiers of Subsurface Energy Security, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under award DE-SC0001114.

http://www.siam.org/journals/mms/12-4/95292.html

†Center for Subsurface Modeling, The Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, Austin, TX 78712 (bganis@ices.utexas.edu, kkumar@ices.utexas.edu, gergina@ices.utexas.edu, mfw@ices.utexas.edu).

‡Corresponding author.

§Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260 (yotov@math.pitt.edu). This author was partially supported by NSF grant DMS 1115856 and DOE grant DE-FG02-04ER25618.

coarse spatially nonconforming subdomains, on which the underlying equations are posed on a local level. These subdomains contain independent discretizations that provide local fine-scale resolution and can be nonmatching on subdomain interfaces. Lagrange multiplier unknowns are introduced on a coarse mortar space, through which appropriate interface conditions are weakly enforced to couple the subdomains on a global level. Figure 1 illustrates a spatial discretization of a global problem into subdomains and interfaces.

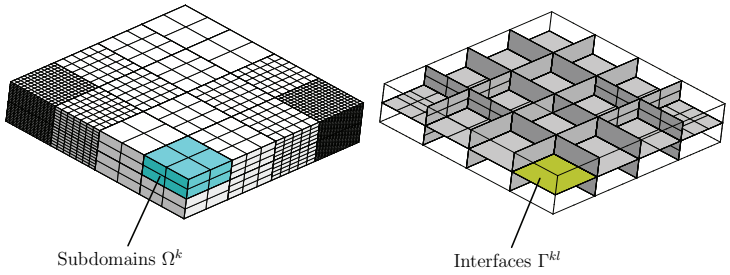

FIG. 1. *Illustration of nonmatching subdomain grids and interfaces.*

Advanced capabilities of the multiscale mortar methods include the ability to model multiple physical models, numerical methods, and discretizations both separately and within the same simulation [38, 3, 42, 46, 45, 31, 27, 4, 18, 39, 35, 17, 41, 15, 14]. These methods have been developed for single-, two-, and three-phase flows in porous media, Stokes–Darcy flows, elasticity and poroelasticity, porescale models, and reactive transport models. In multiphase simulation they have been applied to fully implicit, IMPES, and iteratively coupled schemes. Subdomain discretization methods have included mixed finite elements, multipoint flux approximations, and continuous and discontinuous Galerkin methods on simplices, prisms, quadrilaterals, and hexahedral elements. This work uses mixed finite elements on bricks within the subdomains and considers a two-phase fully implicit model. The use of a mixed method ensures local mass conservation and an accurate representation of fluxes.

Multiscale mortar methods are one member of a large family of multiscale methods in an extensive body of literature. Specific applications to multiphase flows can be found in other methods such as the variational multiscale method [2], hierarchical multiscale method [1], multiscale finite element method [19], and multiscale finite volume method [20]. Many advancements to solvers and preconditioners have been studied, but typically only linear model problems are considered [13]. Effects due to capillarity, gravity, and compressibility are oftentimes neglected in most of the multiscale works on two-phase flow [11, 21, 26]. A notable exception is the recent work of [44], which incorporates capillary pressure and includes upscaling and downscaling of permeabilities in a single domain framework.

This work extends for fully implicit two-phase flow the global Jacobian (GJ) and global Jacobian Schur (GJS) algorithms, which were developed in [14] for slightly compressible single-phase flows. We remark that other global linearization ideas have also recently appeared in the literature in the context of porescale network models [28] and in applications to solid mechanics problems [23]. In this work, a global linearization is performed for both subdomain and interface variables simultaneously to yield a single Newton iteration, resulting in an 8×8 block Jacobian matrix. The four flux variables (and optionally two subdomain unknowns) are eliminated to obtain

linear systems for the 1st (or the 2nd) Schur complements. The 1st Schur complement system has a modular block structure comprising previously available subdomain Jacobian matrices with sparse coupling blocks. The 2nd Schur complement system has a linear interface operator which is solved by a matrix-free approach.

The previous algorithm developed to solve the multiscale mortar method with nonlinear multiphase models was to reduce the global problem to a nonlinear interface problem [45, 46, 31, 15]. Thus two levels of linearizations were employed: one on the interface level and a second on the subdomain level. We refer to this method as the forward difference (FD) algorithm, because of the approximation necessary for the Jacobian on the interface level. See Figure 2 for flow charts comparing the number of nested iterations and tolerances necessary in the GJ, GJS, and FD algorithms. The use of acceleration techniques such as forcing function [12, 10] or preconditioners [45, 46, 15] helped to mitigate the computational expense of the FD method, but the overall complexity of the algorithm still remains.

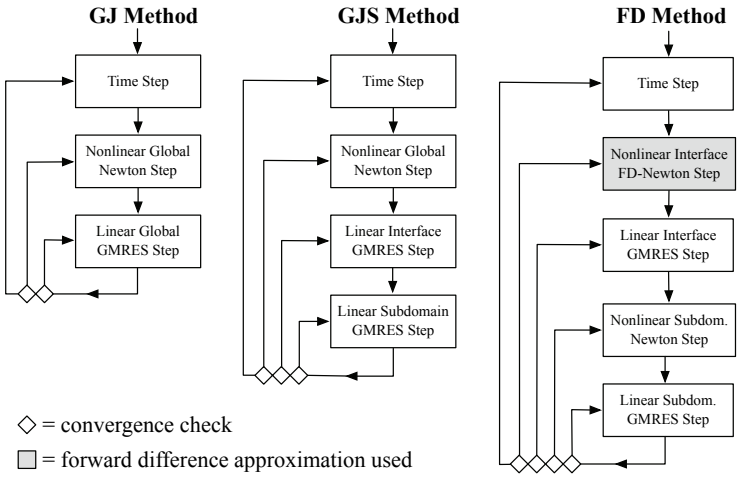

FIG. 2. *Flowcharts for the GJ, GJS, and FD algorithms.*

There are several advantages of the GJ method. The computational cost is observed to be much less expensive than the FD method. The number of global Newton iterations in GJ is observed to be comparable to the number of interface Newton iterations in FD. However, the GJ method does not have the second Newton loop, which is observed to be quite expensive in the FD method. Furthermore, the global linearization provides a simpler algorithm for explicit treatment of nonlinearities without an FD approximation, and it is much more robust by requiring far fewer tolerance parameters (see Figure 2). It also enables use of efficient interface solvers and preconditioners developed for linear interface problems such as multigrid [42], balancing [30], and multiscale flux basis [16]. The modular block structure of the GJ matrix allows this algorithm to use existing reservoir simulators for problems with multiple subdomains and nonmatching grids without much additional effort. From a parallel computing perspective, this method combines the ideas of both domain and data decompositions where the latter is utilized for the linear solver. In the case of a single-phase flow, strong parallel scalability of such a scheme has been shown in [14].

The transport component to the two-phase flow model requires careful integration of phase mobility terms along subdomain interfaces. This has also been observed in

hybrid mixed methods for linear advection-diffusion-reaction problems in [7], where upwind extrapolation schemes have been proposed. In the context of mortars for two-phase flow, we discuss several possibilities for phase mobility upwinding. These are observed to have a significant impact on water saturation accuracy, the ability to achieve nonlinear convergence, and the solvability of linear systems. This is especially important in a global linearization framework where subdomain and mortar variables are tightly coupled. We find the best method to be block-to-block upwinding, which directly projects fluid properties from an adjacent domain. This is a similar idea to the approach used in the enhanced velocity method [37].

The article is organized as follows: In section 2 we provide the equations describing the two-phase model. In section 3 we formulate the multiscale mortar discretization. In section 4 we describe the procedure for global linearization and give the GJ algorithms for the 1st and 2nd Schur complements. In section 5 we discuss different methods for upwinding on subdomain interfaces. In section 6 we present numerical results for a variety of test cases demonstrating efficiency, robustness, and accuracy. In section 7 we give conclusions and future work.

2. Two-phase model. Consider a time interval $[0,T]$, along with a spatial domain, $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, with boundary $\partial\Omega$ and outward unit normal $\boldsymbol{n}$. The spatial domain is decomposed into N_Ω nonoverlapping subdomains such that

$$\overline{\Omega} = \bigcup_{k=1}^{N_\Omega} \overline{\Omega^k}, \qquad \Omega^k \cap \Omega^l = \emptyset \text{ when } k \neq l,$$

where overbars denote set closure. Subdomain interfaces are denoted by $\Gamma^{kl} = \Gamma^{lk} = \partial\Omega^k \cap \partial\Omega^l$, $\Gamma = \bigcup_{1\le k<l\le N_\Omega} \Gamma^{kl}$, and $\boldsymbol{n}^k$ is the outward unit normal to Ω^k.

In this work we consider flow of two immiscible phases through a porous medium, in which each phase $\alpha = \mathrm{o}$ (oil) and $\alpha = \mathrm{w}$ (water) is assumed to be a slightly compressible fluid. In our fully implicit formulation, the primary unknowns will be the oil mass flux $\boldsymbol{u}_\mathrm{o}$, the water mass flux $\boldsymbol{u}_\mathrm{w}$, an auxiliary oil mass flux $\widetilde{\boldsymbol{u}}_\mathrm{o}$, an auxiliary water mass flux $\widetilde{\boldsymbol{u}}_\mathrm{w}$, the oil pressure p_o, the oil concentration n_o, and two Lagrange multipliers λ_1 and λ_2. All remaining unknowns can be expressed in terms of the primary variables: the water pressure p_w, the water concentration n_w, the saturation s_α for each phase, and the fluid density ρ_α for each phase. Known data include the porosity of the medium ϕ, the relative permeability curve $k_{r\alpha}$ for each phase, the fluid viscosity μ_α for each phase, a mass flux source function q_α for each phase, an initial condition $p_{\alpha,0}$ for each phase, and the absolute permeability tensor K. Here K is assumed to be uniformly bounded and positive definite; i.e., there exist constants $0 < k_0 \le k_1 < \infty$ such that

$$k_o\|\xi\|^2 \le \xi^T K(\mathbf{x})\xi \le k_1\|\xi\|^2 \quad \forall \xi \in \mathbb{R}^d, \quad \forall \mathbf{x} \in \Omega, \tag{2.1}$$

where $\|\cdot\|$ is the Euclidean vector norm.

The expanded mixed formulation for fully implicit two-phase flow through a porous medium [46] is given, for subdomains $k = 1, \ldots, N_\Omega$ and phases $\alpha = \mathrm{o}, \mathrm{w}$,

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha) + \nabla \cdot \boldsymbol{u}_\alpha = q_\alpha \qquad \text{in } \Omega^k \times (0,T], \tag{2.2a}$$

$$\widetilde{\boldsymbol{u}}_\alpha = -K(\nabla p_\alpha - \rho_\alpha \boldsymbol{g}) \qquad \text{in } \Omega^k \times (0,T], \tag{2.2b}$$

$$\boldsymbol{u}_\alpha = \frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}\widetilde{\boldsymbol{u}}_\alpha \qquad \text{in } \Omega^k \times (0,T] \tag{2.2c}$$

subject to the initial, boundary, and interface conditions

$$p_\alpha = p_{\alpha,0} \quad \text{at } \Omega \times \{t = 0\}, \tag{2.2d}$$

$$\boldsymbol{u}_\alpha \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega \times (0, T], \tag{2.2e}$$

$$p_\alpha = p_\alpha^\Gamma(\lambda_1, \lambda_2) \quad \text{on } \Gamma \times (0, T], \tag{2.2f}$$

$$\boldsymbol{u}_\alpha^k \cdot \boldsymbol{n}^k + \boldsymbol{u}_\alpha^l \cdot \boldsymbol{n}^l = 0 \quad \text{on } \Gamma^{kl} \times (0, T], \tag{2.2g}$$

with the closure relationships

$$s_{\mathrm{w}} + s_{\mathrm{o}} = 1, \tag{2.2h}$$

$$p_{\mathrm{c}}(s_{\mathrm{w}}) = p_{\mathrm{o}} - p_{\mathrm{w},} \tag{2.2i}$$

$$\rho_\alpha(p_\alpha) = \rho_\alpha^{\mathrm{ref}} \mathrm{e}^{c_\alpha p_\alpha}. \tag{2.2j}$$

For each phase α on each subdomain Ω^i, the subsurface flow is characterized by Darcy's law (2.2b), a conservation of mass (2.2a), and the relationship between phase mass flux to auxiliary mass flux involving phase mobility (2.2c). We assume a hydrostatic initial condition in (2.2d) and no-flow external boundary conditions in (2.2e). The phase pressures on the interfaces are functions of the two Lagrange multipliers in (2.2f), and a conservation of mass is enforced over the entire domain in (2.2g). The saturation constraint (2.2h) describes immiscible flow, the capillary pressure function (2.2i) is a known function of water saturation that relates the two phase pressures, and the equations of state (2.2j) describe the slightly compressible density of each phase with small compressibility constants c_α.

3. Multiscale mortar discretization. The concentration and mobility for phase α is denoted by

$$n_\alpha = \rho_\alpha s_\alpha \quad \text{and} \quad m_\alpha = \frac{k_{r\alpha}\rho_\alpha}{\mu_\alpha}, \tag{3.1}$$

respectively. Note that our definition of mobility includes density, which differs from the literature. The primary unknowns for the multiscale mortar discretization are $(p_{\mathrm{o}}, n_{\mathrm{o}})$, along with velocities $(\widetilde{\boldsymbol{u}}_{\mathrm{o}}, \widetilde{\boldsymbol{u}}_w, \boldsymbol{u}_{\mathrm{o}}, \boldsymbol{u}_{\mathrm{w}})$, and Lagrange multipliers (λ_1, λ_2) corresponding to various physical unknowns on the interface. The functions p_α^Γ in (2.2f) evaluate interface pressure for phase α in terms of the Lagrange multipliers. In this work we use $p_{\mathrm{o}}^\Gamma = \lambda_1$ and $p_{\mathrm{w}}^\Gamma = \lambda_2$, although other possibilities are possible leading to minor modifications in the algorithm.

For each subdomain Ω^k, let $\mathcal{T}_h^k$ be a conforming quasi-uniform finite element partition consisting of rectangular elements or bricks with characteristic mesh size h, which may be different for each k, thereby allowing nonmatching grids on the subdomain interfaces. For the spatial discretization, our formulation can use any of the usual mixed finite element spaces [6], denoted by $\boldsymbol{V}_h^k \times W_h^k$. In particular our numerical results use the lowest order Raviart–Thomas–Nédélec elements [33, 29] on bricks for the flux and the pressure, i.e.,

$$\boldsymbol{V}_h^k = \left\{ \boldsymbol{v} \in H(\mathrm{div}, \Omega^k) : \boldsymbol{v}|_E = (m_1 x_1 + \beta_1, m_2 x_2 + \beta_2, m_3 x_3 + \beta_3)^T, m_l, \beta_l \in \mathbb{R} \right.$$
$$\left. \forall E \in \mathcal{T}_h^k, \text{ and } \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \right\},$$

$$W_h^k = \left\{ w \in L^2(\Omega^k) : w|_E = \gamma, \gamma \in \mathbb{R} \ \forall E \in \mathcal{T}_h^k \right\}.$$

For each interface Γ^{kl}, we choose an independent quasi-uniform finite element partition $\mathcal{G}_H^{kl}$ with characteristic mesh size H and define the mortar space M_H^{kl} containing either continuous or discontinuous piecewise polynomials of degree larger than

one. We remark that for stability of the mortar method, the mortar space cannot be too rich [4].

For each phase α, the global mass flux, pressure, and Lagrange multiplier spaces are

$$\boldsymbol{V}_h = \bigoplus_{k=1}^{N_\Omega} \boldsymbol{V}_h^k, \quad W_h = \bigoplus_{k=1}^{N_\Omega} W_h^k, \quad \text{and} \quad M_H = \bigoplus_{1\le k<l\le N_\Omega} M_H^{kl}.$$

Choose a temporal discretization $0 = t^0 < t^1 < \cdots < t^{N_T} = T$, with $\delta t^n = t^n - t^{n-1}$. To simplify notation we suppress subscripts h and H, as well as time index n except for n_α^{n-1}, which denote the known concentrations at the previous time step. Employing the backward Euler method for time integration with the aforementioned discrete spaces for pressure, mass flux, and mortar variables to system (2.2a)–(2.2j) gives the following fully discrete multiscale mortar expanded mixed finite element system for fully implicit two-phase flow.

(Multiscale mortar expanded mixed finite element method.) For time levels $n = 1, \ldots, N_T$, find $(\widetilde{\boldsymbol{u}}_{\mathrm{o}}^k, \widetilde{\boldsymbol{u}}_w^k, \boldsymbol{u}_{\mathrm{o}}^k, \boldsymbol{u}_{\mathrm{w}}^k, p_{\mathrm{o}}^k, n_{\mathrm{o}}^k, \lambda_1^{kl}, \lambda_2^{kl}) \in 4\boldsymbol{V}_h^k \times 2W_h^k \times 2M_H^{kl}$ such that for subdomains Ω^k, $k = 1, \ldots, N_\Omega$,

$$\int_{\Omega^k} \frac{\phi n_\alpha^k - \phi n_\alpha^{n-1}}{\delta t} w\, dx + \int_{\Omega^k} \nabla \cdot \boldsymbol{u}_\alpha^k w\, dx - \int_{\Omega^k} q_\alpha w\, dx = 0, \tag{3.2a}$$

$$\int_{\Omega^k} K^{-1} \widetilde{\boldsymbol{u}}_\alpha^k \cdot \widetilde{\boldsymbol{v}}\, dx - \int_{\Omega^k} p_\alpha^k \nabla \cdot \widetilde{\boldsymbol{v}}\, dx - \int_{\Omega^k} \rho_\alpha \boldsymbol{g} \cdot \widetilde{\boldsymbol{v}}\, dx + \sum_{l=1, l\neq k}^{N_\Omega} \int_{\Gamma^{kl}} p_\alpha^\Gamma \widetilde{\boldsymbol{v}} \cdot \boldsymbol{n}^k\, d\sigma = 0, \tag{3.2b}$$

$$\int_{\Omega^k} \boldsymbol{u}_\alpha^k \cdot \boldsymbol{v}\, dx - \int_{\Omega^k} m_\alpha \widetilde{\boldsymbol{u}}_\alpha^k \cdot \boldsymbol{v}\, dx = 0, \tag{3.2c}$$

and for interfaces Γ^{kl}, $1 \le k < l \le N_\Omega$,

$$\int_{\Gamma^{kl}} \left(\boldsymbol{u}_\alpha^k \cdot \boldsymbol{n}^k + \boldsymbol{u}_\alpha^l \cdot \boldsymbol{n}^l \right) \eta\, d\sigma = 0, \tag{3.2d}$$

for all $w \in W_h^k$, $\widetilde{\boldsymbol{v}} \in \boldsymbol{V}_h^k$, $\boldsymbol{v} \in \boldsymbol{V}_h^k$, and $\eta \in M_H^{kl}$. Note that p_w^k, ρ_o^k, ρ_w^k, n_w^k, m_w^k, and m_o^k are all functions of primary unknowns (p_o^k, n_o^k) using (2.2h)–(2.2j).

At this point, our two-phase multiscale mortar formulation will depart from the work in the previous literature. Let $\{\boldsymbol{v}_i^k\}_{i=1}^{N_{\boldsymbol{u}}^k}, \{w_i^k\}_{i=1}^{N_p^k}, \{\eta_i^{kl}\}_{i=1}^{N_\lambda^{kl}}$ be finite element basis functions for V_h^k, W_h^k, and M_H^{kl}, respectively. The 8 unknowns are then expressed as linear combinations of the basis functions

$$\begin{aligned}
&\widetilde{\boldsymbol{u}}_{\mathrm{o}}^k = \sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{o},i}^k \boldsymbol{v}_i^k, \quad \widetilde{\boldsymbol{u}}_{\mathrm{w}}^k = \sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{w},i}^k \boldsymbol{v}_i^k, \quad \boldsymbol{u}_{\mathrm{o}}^k = \sum_{i=1}^{N_{\boldsymbol{u}}^k} U_{\mathrm{o},i}^k \boldsymbol{v}_i^k, \quad \boldsymbol{u}_{\mathrm{w}}^k = \sum_{i=1}^{N_{\boldsymbol{u}}^k} U_{\mathrm{w},i}^k \boldsymbol{v}_i^k, \\
&p_{\mathrm{o}}^k = \sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k, \quad n_{\mathrm{o}}^k = \sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k, \quad \lambda_1^{kl} = \sum_{i=1}^{N_\lambda^{kl}} \Lambda_{1,i}^{kl} \eta_i^{kl}, \quad \lambda_2^{kl} = \sum_{i=1}^{N_\lambda^{kl}} \Lambda_{2,i}^{kl} \eta_i^{kl},
\end{aligned} \tag{3.3}$$

and the coefficient vectors $\widetilde{U}_{\mathrm{o}}^k, \widetilde{U}_{\mathrm{w}}^k, U_{\mathrm{o}}^k, U_{\mathrm{w}}^k \in \mathbb{R}^{N_{\boldsymbol{u}}^k}$, $P_{\mathrm{o}}^k, N_{\mathrm{o}}^k \in \mathbb{R}^{N_p^k}$, and $\Lambda_1^{kl}, \Lambda_2^{kl} \in \mathbb{R}^{N_\lambda^{kl}}$ are to be determined. These are grouped together by subdomains and interfaces

to form the global coefficient vector unknowns. Defining variables for the total number of degrees of freedom for the mass flux, pressure, and Lagrange multiplier,

$$N_{\boldsymbol{u}} = \sum_{i=1}^{N_\Omega} N_{\boldsymbol{u}}^k, \quad N_p = \sum_{i=1}^{N_\Omega} N_p^k, \quad \text{and} \quad N_\lambda = \sum_{1\le k<l\le N_\Omega} N_\lambda^{kl},$$

the global coefficient vectors are $\widetilde{U}_{\mathrm{o}}, \widetilde{U}_{\mathrm{w}}, U_{\mathrm{o}}, U_{\mathrm{w}} \in \mathbb{R}^{N_{\boldsymbol{u}}}$, $P_{\mathrm{o}}, N_{\mathrm{o}} \in \mathbb{R}^{N_p}$, and $\Lambda_1, \Lambda_2 \in \mathbb{R}^{N_\lambda}$. Substituting the representations from (3.3) into the variational formulation (3.2a)–(3.2d), we obtain the following equations in residual form:

(Auxiliary equations.) For $k = 1, \dots, N_\Omega$ and $j = 1, \dots, N_{\boldsymbol{u}}^k$,

(3.4a)
$$A_{\mathrm{o},j}^k = \int_{\Omega^k} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U_{\mathrm{o},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx - \int_{\Omega^k} m_{\mathrm{o}} \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k, \sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k \right) \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{o},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx,$$

(3.4b)
$$A_{\mathrm{w},j}^k = \int_{\Omega^k} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U_{\mathrm{w},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx - \int_{\Omega^k} m_{\mathrm{w}} \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k, \sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k \right) \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{w},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx.$$

(Darcy law equations.) For $k = 1, \dots, N_\Omega$ and $j = 1, \dots, N_{\boldsymbol{u}}^k$,

(3.4c)
$$\begin{aligned} D_{\mathrm{o},j}^k = {} & \int_{\Omega^k} K^{-1} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{o},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx - \int_{\Omega^k} \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k \right) \nabla \cdot \boldsymbol{v}_j^k \, dx \\ & - \int_{\Omega^k} \rho_{\mathrm{o}} \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k \right) \mathbf{g} \cdot \boldsymbol{v}_j^k \, dx + \sum_{l=1, l\neq k}^{N_\Omega} \int_{\Gamma^{kl}} p_{\mathrm{o}}^\Gamma \left(\sum_{i=1}^{N_\lambda^k} \Lambda_{1,i}^{kl} \eta_i^{kl}, \sum_{i=1}^{N_\lambda^k} \Lambda_{2,i}^{kl} \eta_i^{kl} \right) \boldsymbol{v}_j^k \cdot \boldsymbol{n}^k \, d\sigma, \end{aligned}$$

(3.4d)
$$\begin{aligned} D_{\mathrm{w},j}^k = {} & \int_{\Omega^k} K^{-1} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} \widetilde{U}_{\mathrm{w},i}^k \boldsymbol{v}_i^k \right) \cdot \boldsymbol{v}_j^k \, dx - \int_{\Omega^k} p_w \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k, \sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k \right) \nabla \cdot \boldsymbol{v}_j^k \, dx \\ & - \int_{\Omega^k} \rho_{\mathrm{w}} \left(\sum_{i=1}^{N_p^k} P_{\mathrm{o},i}^k w_i^k, \sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k \right) \mathbf{g} \cdot \boldsymbol{v}_j^k \, dx \\ & + \sum_{l=1, l\neq k}^{N_\Omega} \int_{\Gamma^{kl}} p_{\mathrm{w}}^\Gamma \left(\sum_{i=1}^{N_\lambda^k} \Lambda_{1,i}^{kl} \eta_i^{kl}, \sum_{i=1}^{N_\lambda^k} \Lambda_{2,i}^{kl} \eta_i^{kl} \right) \boldsymbol{v}_j^k \cdot \boldsymbol{n}^k \, d\sigma. \end{aligned}$$

(Mass balance equations.) For $k = 1, \dots, N_\Omega$ and $j = 1, \dots, N_p^k$,

(3.4e)
$$B_{\mathrm{o},j}^k = \int_{\Omega^k} \phi \left[\sum_{i=1}^{N_p^k} N_{\mathrm{o},i}^k w_i^k - n_{\mathrm{o}}^{n-1} \right] w_j^k \, dx + \delta t \int_{\Omega^k} \left[\nabla \cdot \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U_{\mathrm{o},i}^k \boldsymbol{v}_i^k \right) - q_{\mathrm{o}} \right] w_j^k \, dx,$$

(3.4f)

$$B^k_{\mathrm{w},j} = \int_{\Omega^k} \phi \left[n_w \left(\sum_{i=1}^{N_p^k} P^k_{\mathrm{o},i} w^k_i, \sum_{i=1}^{N_p^k} N^k_{\mathrm{o},i} w^k_i \right) - n_{\mathrm{w}}^{n-1} \right] w^k_j \, dx + \delta t \int_{\Omega^k} \left[\nabla \cdot \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U^k_{\mathrm{w},i} \boldsymbol{v}^k_i \right) - q_{\mathrm{w}} \right] w^k_j \, dx.$$

(Flux continuity equations.) For $k = 1, \ldots, N_\Omega - 1$, $l = (k+1), \ldots, N_\Omega$, and $j = 1, \ldots, N^{kl}_\lambda$,

$$H^{kl}_{\mathrm{o},j} = \int_{\Gamma^{kl}} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U^k_{\mathrm{o},i} \boldsymbol{v}^k_i \cdot \boldsymbol{n}^k + \sum_{i=1}^{N_{\boldsymbol{u}}^l} U^l_{\mathrm{o},i} \boldsymbol{v}^l_i \cdot \boldsymbol{n}^l \right) \eta^{kl}_j \, d\sigma, \tag{3.4g}$$

$$H^{kl}_{\mathrm{w},j} = \int_{\Gamma^{kl}} \left(\sum_{i=1}^{N_{\boldsymbol{u}}^k} U^k_{\mathrm{w},i} \boldsymbol{v}^k_i \cdot \boldsymbol{n}^k + \sum_{i=1}^{N_{\boldsymbol{u}}^l} U^l_{\mathrm{w},i} \boldsymbol{v}^l_i \cdot \boldsymbol{n}^l \right) \eta^{kl}_j \, d\sigma. \tag{3.4h}$$

Here we have used the closure relationships (2.2h), (2.2i), and (2.2j) to express the functions $p_{\mathrm{w}} = p_{\mathrm{w}}(P_{\mathrm{o}}, N_{\mathrm{o}})$, $\rho_{\mathrm{o}} = \rho_{\mathrm{o}}(P_{\mathrm{o}})$, $\rho_{\mathrm{w}} = \rho_{\mathrm{w}}(P_{\mathrm{o}}, N_{\mathrm{o}})$, $n_{\mathrm{w}} = n_{\mathrm{w}}(P_{\mathrm{o}}, N_{\mathrm{o}})$, $m_{\mathrm{w}} = m_{\mathrm{w}}(P_{\mathrm{o}}, N_{\mathrm{o}})$, and $m_{\mathrm{o}} = m_{\mathrm{o}}(P_o, N_o)$ in terms of the unknown coefficients P_{o} and N_{o}.

4. Global linearization. The residual equations constitute a set of $(4N_{\boldsymbol{u}} + 2N_p + 2N_\lambda)$ nonlinear equations which will be solved by a Newton method, where $(\widetilde{U}_{\mathrm{o}}, \widetilde{U}_{\mathrm{w}}, U_{\mathrm{o}}, U_{\mathrm{w}}, P_{\mathrm{o}}, N_{\mathrm{o}}, \Lambda_1, \Lambda_2)$ is the unknown coefficient vector before elimination. The system takes the following form:

$$\begin{aligned} A_{\mathrm{o}}(\widetilde{U}_{\mathrm{o}}, U_{\mathrm{o}}, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0, & B_{\mathrm{o}}(U_o, N_{\mathrm{o}}) &= 0, \\ A_{\mathrm{w}}(\widetilde{U}_{\mathrm{w}}, U_{\mathrm{w}}, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0, & B_{\mathrm{w}}(U_w, P_{\mathrm{o}}, N_{\mathrm{o}}) &= 0, \\ D_{\mathrm{o}}(\widetilde{U}_{\mathrm{o}}, P_{\mathrm{o}}, \Lambda_1, \Lambda_2) &= 0, & H_{\mathrm{o}}(U_{\mathrm{o}}) &= 0, \\ D_{\mathrm{w}}(\widetilde{U}_{\mathrm{w}}, P_{\mathrm{o}}, N_{\mathrm{o}}, \Lambda_1, \Lambda_2) &= 0, & H_{\mathrm{w}}(U_{\mathrm{w}}) &= 0. \end{aligned} \tag{4.1}$$

To form the Jacobian matrix, we compute the derivatives of each residual equation with respect to each unknown. Terms that will be simplified are denoted with hats. We denote the matrices:

$$(A^k_1)_{ji} = \frac{\partial A^k_{\mathrm{o},j}}{\partial \widetilde{U}^k_{\mathrm{o},i}}, \quad (A^k_2)_{ji} = \frac{\partial A^k_{\mathrm{o},j}}{\partial U^k_{\mathrm{o},i}}, \quad (\widehat{A}^k_3)_{ji} = \frac{\partial A^k_{\mathrm{o},j}}{\partial P^k_{\mathrm{o},i}}, \quad (A^k_4)_{ji} = \frac{\partial A^k_{\mathrm{o},j}}{\partial N^k_{\mathrm{o},i}},$$

$$(B^k_1)_{ji} = \frac{\partial A^k_{\mathrm{w},j}}{\partial \widetilde{U}^k_{\mathrm{w},i}}, \quad (B^k_2)_{ji} = \frac{\partial A^k_{\mathrm{w},j}}{\partial U^k_{\mathrm{w},i}}, \quad (\widehat{B}^k_3)_{ji} = \frac{\partial A^k_{\mathrm{w},j}}{\partial P^k_{\mathrm{o},i}}, \quad (\widehat{B}^k_4)_{ji} = \frac{\partial A^k_{\mathrm{w},j}}{\partial N^k_{\mathrm{o},i}},$$

$$(C^k_1)_{ji} = \frac{\partial D^k_{\mathrm{o},j}}{\partial \widetilde{U}^k_{\mathrm{o},i}}, \quad (\widehat{C}^k_2)_{ji} = \frac{\partial D^k_{\mathrm{o},j}}{\partial P^k_{\mathrm{o},i}}, \quad (C^{kl}_3)_{ji} = \frac{\partial D^k_{\mathrm{o},j}}{\partial \Lambda^{kl}_{1,i}}, \quad (C^{kl}_4)_{ji} = \frac{\partial D^k_{\mathrm{o},j}}{\partial \Lambda^{kl}_{2,i}},$$

$$(D^k_1)_{ji} = \frac{\partial D^k_{\mathrm{w},j}}{\partial \widetilde{U}^k_{\mathrm{w},i}}, \quad (\widehat{D}^k_2)_{ji} = \frac{\partial D^k_{\mathrm{w},j}}{\partial P^k_{\mathrm{o},i}}, \quad (\widehat{D}^k_3)_{ji} = \frac{\partial D^k_{\mathrm{w},j}}{\partial N^k_{\mathrm{o},i}}, \quad (D^{kl}_4)_{ji} = \frac{\partial D^k_{\mathrm{w},j}}{\partial \Lambda^{kl}_{1,i}},$$

$$(D^{kl}_5)_{ji} = \frac{\partial D^k_{\mathrm{w},j}}{\partial \Lambda^{kl}_{2,i}},$$

$$(E^k_1)_{ji} = \frac{\partial B^k_{\mathrm{o},j}}{\partial U^k_{\mathrm{o},i}}, \quad (E^k_2)_{ji} = \frac{\partial B^k_{\mathrm{o},j}}{\partial N^k_{\mathrm{o},i}},$$

$$(F_1^k)_{ji} = \frac{\partial B^k_{\mathrm{w},j}}{\partial U^k_{\mathrm{w},i}}, \quad (\widehat{F}_2^k)_{ji} = \frac{\partial B^k_{\mathrm{w},j}}{\partial P^k_{\mathrm{o},i}}, \quad (\widehat{F}_3^k)_{ji} = \frac{\partial B^k_{\mathrm{w},j}}{\partial N^k_{\mathrm{o},i}},$$
$$(L_1^{kl})_{ji} = \frac{\partial H^{kl}_{\mathrm{o},j}}{\partial U^k_{\mathrm{o},i}},$$
$$(L_2^{kl})_{ji} = \frac{\partial H^{kl}_{\mathrm{w},j}}{\partial U^k_{\mathrm{w},i}}.$$

Let $(\cdot,\cdot)_k$ denote the L^2 inner product over subdomain Ω^k, and let $\langle\cdot,\cdot\rangle_{kl}$ denote the L^2 inner product over interface Γ^{kl}. We use the residual formulation (3.4a)–(3.4h) to compute the following partial derivatives, omitting superscript k in the test functions to simplify notation:

$$(A_1^k)_{ji} = -\left(m_{\mathrm{o}}\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2a}$$
$$(A_2^k)_{ji} = \left(\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2b}$$
$$(\widehat{A}_3^k)_{ji} = -\left(c_{\mathrm{o}}\left(\frac{n_{\mathrm{o}}}{\mu_{\mathrm{o}}}k'_{r\mathrm{o}} + \frac{\rho_{\mathrm{o}}}{\mu_o}k_{r\mathrm{o}}\right) w_i\widetilde{\boldsymbol{u}}_{\mathrm{o}}, \boldsymbol{v}_j\right)_k, \tag{4.2c}$$
$$(A_4^k)_{ji} = -\left(\frac{1}{\mu_{\mathrm{o}}}k'_{r\mathrm{o}}w_i\widetilde{\boldsymbol{u}}_{\mathrm{o}}, \boldsymbol{v}_j\right)_k, \tag{4.2d}$$
$$(B_1^k)_{ji} = -\left(m_{\mathrm{w}}\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2e}$$
$$(B_2^k)_{ji} = \left(\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2f}$$
$$(\widehat{B}_3^k)_{ji} = -\left(\frac{\rho_{\mathrm{w}}}{\mu_{\mathrm{w}}}\left(c_{\mathrm{o}}\frac{n_{\mathrm{o}}}{\rho_{\mathrm{o}}}k'_{r\mathrm{w}} + c_{\mathrm{w}}k_{r\mathrm{w}}\right) w_i\widetilde{\boldsymbol{u}}_{\mathrm{w}}, \boldsymbol{v}_j\right)_k, \tag{4.2g}$$
$$(\widehat{B}_4^k)_{ji} = -\left(\frac{\rho_w}{\mu_{\mathrm{w}}\rho_{\mathrm{o}}}\left(c_{\mathrm{w}}k_{r\mathrm{w}}p'_c - k'_{r\mathrm{w}}\right)\, w_i\,\widetilde{\boldsymbol{u}}_{\mathrm{w}}, \boldsymbol{v}_j\right)_k, \tag{4.2h}$$
$$(C_1^k)_{ji} = \left(K^{-1}\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2i}$$
$$(\widehat{C}_2^k)_{ji} = -\left(w_i, \nabla\cdot\boldsymbol{v}_j\right)_k - \left(c_{\mathrm{o}}\rho_{\mathrm{o}}\boldsymbol{g}w_i, \boldsymbol{v}_j\right)_k, \tag{4.2j}$$
$$(C_3^{kl})_{ji} = \left\langle\eta_i^{kl}, \boldsymbol{v}_j^k\cdot\boldsymbol{n}^k\right\rangle_{kl}, \tag{4.2k}$$
$$(C_4^{kl})_{ji} = 0, \tag{4.2l}$$
$$(D_1^k)_{ji} = \left(K^{-1}\boldsymbol{v}_i, \boldsymbol{v}_j\right)_k, \tag{4.2m}$$
$$(\widehat{D}_2^k)_{ji} = -\left(\left(1 - \frac{c_{\mathrm{o}}n_{\mathrm{o}}}{\rho_{\mathrm{o}}}p'_c\right) w_i, \nabla\cdot\boldsymbol{v}_j\right)_k - \left(c_{\mathrm{w}}\rho_{\mathrm{w}}\left(1 - c_{\mathrm{o}}\frac{n_{\mathrm{o}}p'_c}{\rho_{\mathrm{o}}}\right)\boldsymbol{g}w_i, \boldsymbol{v}_j\right)_k, \tag{4.2n}$$
$$(\widehat{D}_3^k)_{ji} = -\left(\frac{1}{\rho_{\mathrm{o}}}p'_c w_i, \nabla\cdot\boldsymbol{v}_j\right)_k - \left(\frac{c_{\mathrm{w}}\rho_{\mathrm{w}}}{\rho_{\mathrm{o}}}p'_c\boldsymbol{g}w_i, \boldsymbol{v}_j\right)_k, \tag{4.2o}$$
$$(D_4^{kl})_{ji} = 0, \tag{4.2p}$$
$$(D_5^{kl})_{ji} = \left\langle\eta_i^{kl}, \boldsymbol{v}_j^k\cdot\boldsymbol{n}^k\right\rangle_{kl}, \tag{4.2q}$$
$$(E_1^k)_{ji} = \delta t\left(\nabla\cdot\boldsymbol{v}_i, w_j\right)_k, \tag{4.2r}$$
$$(E_2^k)_{ji} = \left(\phi w_i, w_j\right)_k, \tag{4.2s}$$
$$(F_1^k)_{ji} = \delta t\left(\nabla\cdot\boldsymbol{v}_i, w_j\right)_k, \tag{4.2t}$$
$$(\widehat{F}_2^k)_{ji} = \left(\phi\left(c_{\mathrm{w}}n_{\mathrm{w}} + c_{\mathrm{o}}\frac{\rho_{\mathrm{w}}n_{\mathrm{o}}}{\rho_{\mathrm{o}}}\left(1 - c_{\mathrm{w}}p'_c\right)\right) w_i, w_j\right)_k, \tag{4.2u}$$

$$(\widehat{F}_3^k)_{ji} = \left(\frac{\phi}{\rho_{\mathrm{o}}} \left(c_{\mathrm{w}} n_{\mathrm{w}} p_c' - \rho_{\mathrm{w}} \right) w_i, w_j \right)_k , \tag{4.2v}$$

$$(L_1^{kl})_{ji} = \langle \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k, \eta_j^{kl} \rangle_{kl}, \tag{4.2w}$$

$$(L_2^{kl})_{ji} = \langle \boldsymbol{v}_i^k \cdot \boldsymbol{n}^k, \eta_j^{kl} \rangle_{kl}. \tag{4.2x}$$

Entries with two pressure basis functions are evaluated using the midpoint rule, and entries with one or more velocity basis functions are evaluated using the trapezoidal midpoint rule as in [5]. Entries with mortar basis functions are evaluated using composite Newton–Cotes rules with accuracy depending on the polynomial degree of the mortar space. For example, an interface with a quadratic mortar requires the composite Simpson's rule. Mobilities m_o and m_w are upwinded and will be discussed in section 5.

Note that in the above computations the Lagrange multipliers have the meaning of phase pressures at the interfaces, which explains the zero contribution for C_4 and D_4. The formulation allows for a different choice such as n_o, p_o, in which case the block matrices C_3, C_4, D_4, D_5 will be different.

4.1. Jacobian simplification and global assembly. We simplify the above Jacobian using the technique that is typically performed for slightly compressible flow models, by which we drop certain terms that are multiplied by the small compressibility constants c_{o} and c_{w}. Note that these terms are small and would be exactly zero in the case of incompressible fluids. Specifically we take

$$(\widehat{A}_3^k)_{ji} \approx 0, \tag{4.3a}$$

$$(\widehat{B}_3^k)_{ji} \approx 0, \tag{4.3b}$$

$$(\widehat{B}_4^k)_{ji} \approx (B_4^k)_{ji} = \left(\frac{\rho_w}{\mu_{\mathrm{w}} \rho_{\mathrm{o}}} k_{r\mathrm{w}}' \, w_i \, \widetilde{\boldsymbol{u}}_{\mathrm{w}}, \boldsymbol{v}_j \right)_k , \tag{4.3c}$$

$$(\widehat{C}_2^k)_{ji} \approx (C_2^k)_{ji} = - \left(w_i, \nabla \cdot \boldsymbol{v}_j \right)_k , \tag{4.3d}$$

$$(\widehat{D}_2^k)_{ji} \approx (D_2^k)_{ji} = - \left(w_i, \nabla \cdot \boldsymbol{v}_j \right)_k , \tag{4.3e}$$

$$(\widehat{D}_3^k)_{ji} \approx (D_3^k)_{ji} = - \left(\frac{1}{\rho_{\mathrm{o}}} p_c' w_i, \nabla \cdot \boldsymbol{v}_j \right)_k , \tag{4.3f}$$

$$(\widehat{F}_2^k)_{ji} \approx (F_2^k)_{ji} = \left(\phi \rho_w \left(c_{\mathrm{w}} s_{\mathrm{w}} + c_{\mathrm{o}} s_{\mathrm{o}} \right) w_i, w_j \right)_k , \tag{4.3g}$$

$$(\widehat{F}_3^k)_{ji} \approx (F_3^k)_{ji} = - \left(\phi \frac{\rho_{\mathrm{w}}}{\rho_{\mathrm{o}}} w_i, w_j \right)_k . \tag{4.3h}$$

This yields an approximation to the Jacobian (matrix), but the residual vector (right-hand side) is still computed exactly. Consistent with the theory on inexact Newton methods, when convergence is achieved, the solution to this nonlinear system coincides with that of the full Newton method [22]. With the above derivations and simplifications complete, we note the following identities: $A_2 = B_2$, $C_1 = D_1$, $C_2 = D_2$, $E_1 = F_1$, and $L_1 = L_2$.

To form the global matrix, subdomain matrices are grouped together as block diagonal matrices, and interface matrices are grouped together by interface. For example, matrices A_1, C_3, and L_1 are defined as

$$A_1 = \begin{pmatrix} A_1^1 & & \\ & \ddots & \\ & & A_1^{N_\Omega} \end{pmatrix}, \; C_3 = \begin{pmatrix} C_3^{12} \\ \vdots \\ C_3^{(N_\Omega - 1) N_\Omega} \end{pmatrix}, \quad \text{and } L_1 = \begin{pmatrix} L_1^{12} & \cdots & L_1^{(N_\Omega - 1) N_\Omega} \end{pmatrix}.$$

Without loss of generality, we have assumed that Γ^{12} and $\Gamma^{(N_\Omega-1)N_\Omega}$ are nonempty. Using the above-defined block matrices, the linear system for the inexact Newton step takes the form

$$\begin{bmatrix} A_1 & 0 & A_2 & 0 & 0 & A_4 & 0 & 0 \\ 0 & B_1 & 0 & B_2 & 0 & B_4 & 0 & 0 \\ C_1 & 0 & 0 & 0 & C_2 & 0 & C_3 & C_4 \\ 0 & D_1 & 0 & 0 & D_2 & D_3 & D_4 & D_5 \\ 0 & 0 & E_1 & 0 & 0 & E_2 & 0 & 0 \\ 0 & 0 & 0 & F_1 & F_2 & F_3 & 0 & 0 \\ 0 & 0 & L_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_2 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\widetilde{U}_{\mathrm{o}} \\ \delta\widetilde{U}_{\mathrm{w}} \\ \delta U_{\mathrm{o}} \\ \delta U_{\mathrm{w}} \\ \delta P_{\mathrm{o}} \\ \delta N_{\mathrm{o}} \\ \delta\Lambda_1 \\ \delta\Lambda_2 \end{bmatrix} = - \begin{bmatrix} A_{\mathrm{o}} \\ A_{\mathrm{w}} \\ D_{\mathrm{o}} \\ D_{\mathrm{w}} \\ B_{\mathrm{o}} \\ B_{\mathrm{w}} \\ H_{\mathrm{o}} \\ H_{\mathrm{w}} \end{bmatrix}. \tag{4.4}$$

4.2. Velocity elimination. We first eliminate both primary and auxiliary mass fluxes for each fluid phase from system (4.4) to form a Schur complement system with the unknowns

$$\delta\Theta = \begin{bmatrix} \delta P_{\mathrm{o}} \\ \delta N_{\mathrm{o}} \end{bmatrix} \quad \text{and} \quad \delta\Lambda = \begin{bmatrix} \delta\Lambda_1 \\ \delta\Lambda_2 \end{bmatrix}. \tag{4.5a}$$

This is accomplished by partial Gaussian elimination. The reduced system has the block structure

$$J \begin{bmatrix} \delta\Theta \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} J_{\Theta\Theta} & J_{\Theta\Lambda} \\ J_{\Lambda\Theta} & J_{\Lambda\Lambda} \end{bmatrix} \begin{bmatrix} \delta\Theta \\ \delta\Lambda \end{bmatrix} = \begin{bmatrix} R_\Theta \\ R_\Lambda \end{bmatrix}, \tag{4.5b}$$

with the matrices

$$J_{\Theta\Theta} = \begin{bmatrix} J_{P_{\mathrm{o}}P_{\mathrm{o}}} & J_{P_{\mathrm{o}}N_{\mathrm{o}}} \\ J_{N_{\mathrm{o}}P_{\mathrm{o}}} & J_{N_{\mathrm{o}}N_{\mathrm{o}}} \end{bmatrix} = \begin{bmatrix} E_1A_2^{-1}A_1C_1^{-1}C_2 & E_2 - E_1A_2^{-1}A_4 \\ F_2 + F_1B_2^{-1}B_1D_1^{-1}D_2 & F_3 + F_1B_2^{-1}(B_1D_1^{-1}D_3 - B_4) \end{bmatrix}, \tag{4.5c}$$

$$J_{\Theta\Lambda} = \begin{bmatrix} J_{P_{\mathrm{o}}\Lambda_1} & J_{P_{\mathrm{o}}\Lambda_2} \\ J_{N_{\mathrm{o}}\Lambda_1} & J_{N_{\mathrm{o}}\Lambda_2} \end{bmatrix} = \begin{bmatrix} E_1A_2^{-1}A_1C_1^{-1}C_3 & E_1A_2^{-1}A_1C_1^{-1}C_4 \\ F_1B_2^{-1}B_1D_1^{-1}D_4 & F_1B_2^{-1}B_1D_1^{-1}D_5 \end{bmatrix}, \tag{4.5d}$$

$$J_{\Lambda\Theta} = \begin{bmatrix} J_{\Lambda_1P_{\mathrm{o}}} & J_{\Lambda_1N_{\mathrm{o}}} \\ J_{\Lambda_2P_{\mathrm{o}}} & J_{\Lambda_2N_{\mathrm{o}}} \end{bmatrix} = \begin{bmatrix} L_1A_2^{-1}A_1C_1^{-1}C_2 & -L_1A_2^{-1}A_4 \\ L_2B_2^{-1}B_1D_1^{-1}D_2 & L_2B_2^{-1}(B_1D_1^{-1}D_3 - B_4) \end{bmatrix}, \tag{4.5e}$$

$$J_{\Lambda\Lambda} = \begin{bmatrix} J_{\Lambda_1\Lambda_1} & J_{\Lambda_1\Lambda_2} \\ J_{\Lambda_2\Lambda_1} & J_{\Lambda_2\Lambda_2} \end{bmatrix} = \begin{bmatrix} L_1A_2^{-1}A_1C_1^{-1}C_3 & L_1A_2^{-1}A_1C_1^{-1}C_4 \\ L_2B_2^{-1}B_1D_1^{-1}D_4 & L_2B_2^{-1}B_1D_1^{-1}D_5 \end{bmatrix}, \tag{4.5f}$$

and the residual vectors

$$R_\Theta = \begin{bmatrix} R_{P_{\mathrm{o}}} \\ R_{N_{\mathrm{o}}} \end{bmatrix} = \begin{bmatrix} E_1A_2^{-1}(A_{\mathrm{o}} - A_1C_1^{-1}D_{\mathrm{o}}) - B_{\mathrm{o}} \\ F_1B_2^{-1}(A_{\mathrm{w}} - B_1D_1^{-1}D_{\mathrm{w}}) - B_{\mathrm{w}} \end{bmatrix}, \tag{4.5g}$$

$$R_\Lambda = \begin{bmatrix} R_{\Lambda_1} \\ R_{\Lambda_2} \end{bmatrix} = \begin{bmatrix} L_1A_2^{-1}(A_{\mathrm{o}} - A_1C_1^{-1}D_{\mathrm{o}}) - H_{\mathrm{o}} \\ L_2B_2^{-1}(A_{\mathrm{w}} - B_1D_1^{-1}D_{\mathrm{w}}) - H_{\mathrm{w}} \end{bmatrix}. \tag{4.5h}$$

The GJ method (Algorithm 1), which recently appeared in [14] for slightly compressible single-phase flow, can now be directly applied to the fully implicit two-phase system (4.5b) with two Lagrange multipliers. Note that in practice the entries in

Algorithm 1. The GJ method.

Given $[\Theta^0, \Lambda^0]$,
for $n = 1, \ldots, N_T$ **do**
  $[\Theta^{n,0}, \Lambda^{n,0}] = [\Theta^{n-1}, \Lambda^{n-1}]$
  for $k = 0, \ldots,$ NEWT_MAX **do**
    $R^{n,k} = [R_\Theta, R_\Lambda](\Theta^{n,k}, \Lambda^{n,k})$
    if ($\|R^{n,k}\| <$ NEWT_TOL) **then break** k-loop
    $J^{n,k} = [J_{\Theta\Theta}, J_{\Theta\Lambda}; J_{\Lambda\Theta}, J_{\Lambda\Lambda}](\Theta^{n,k}, \Lambda^{n,k})$
    $[\delta\Theta, \delta\Lambda] =$ **gmres**$(J^{n,k}, R^{n,k})$
    $[\Theta^{n,k+1}, \Lambda^{n,k+1}] = [\Theta^{n,k} + \delta\Theta, \Lambda^{n,k} + \delta\Lambda]$
  end for
  $[\Theta^n, \Lambda^n] = [\Theta^{n,k}, \Lambda^{n,k}]$
end for

(4.5c)–(4.5h) reduce to simple expressions after quadrature, which can be very efficiently implemented as a cell-centered finite difference code without matrix-matrix or matrix-vector products. Since the matrix J is highly nonsymmetric, the generalized minimum residual (GMRES) method [22] can be used to iteratively solve the linear system on each Newton step. We note that GMRES may be replaced with another suitable linear solver.

4.3. Interface formulation. We can also eliminate subdomain primary variables from system (4.5b) to form a second Schur complement system with $\delta\Lambda = (\delta\Lambda_1, \delta\Lambda_2)^T$ as the sole unknowns. This linear interface problem has the form

$$J_S \; \delta\Lambda = R_S, \tag{4.6a}$$

where

$$J_S = J_{\Lambda\Lambda} - J_{\Lambda\Theta} J_{\Theta\Theta}^{-1} J_{\Theta\Lambda} \qquad \text{and} \tag{4.6b}$$

$$R_S = R_\Lambda - J_{\Lambda\Theta} J_{\Theta\Theta}^{-1} R_\Theta. \tag{4.6c}$$

Note that $J_{\Theta\Theta}$ is a block diagonal matrix, and each decoupled block of subdomain unknowns is invertible. The matrix J_S need not be formed, and its action is obtained in the solution of successive linear subdomain problems. We refer to this strategy as the global Jacobian Schur (GJS) method given by Algorithm 2. This enables the use of previously developed preconditioners and techniques for the linear interface problem such as multigrid [42], balancing [30], and construction of a multiscale flux basis [16]. We note again that GMRES may be replaced with another suitable linear solver.

The GJS algorithm gives a much simpler domain decomposition strategy for nonlinear model problems, because the interface problem has already been linearized. We remark that previous attempts at developing fast solvers for nonlinear interface problems have included Neumann–Neumann preconditioners, nonlinear multigrid methods, and the frozen Jacobian multiscale preconditioner [46, 45, 15]. These methods showed moderate success at reducing computation costs, but the implementations were very complex.

5. Upwinding mobilities. The numerical integration of mobility terms (4.2a) and (4.2e) is essential to capturing accurate transport of phase saturation. Within

Algorithm 2. The GJS method.

Given $[\Theta^0, \Lambda^0]$,
for $n = 1, \ldots, N_T$ **do**
  $[\Theta^{n,0}, \Lambda^{n,0}] = [\Theta^{n-1}, \Lambda^{n-1}]$
  for $k = 0, \ldots,$ NEWT_MAX **do**
    $R^{n,k} = [R_\Theta, R_\Lambda](\Theta^{n,k}, \Lambda^{n,k})$
    if ($\|R^{n,k}\| <$ NEWT_TOL) **then break** k-loop
    $J^{n,k} = [J_{\Theta\Theta}, J_{\Theta\Lambda}; J_{\Lambda\Theta}, J_{\Lambda\Lambda}](\Theta^{n,k}, \Lambda^{n,k})$
    function $J_S(\xi)$ {**return** $J_{\Lambda\Lambda}\xi - J_{\Lambda\Theta}$ **gmres**$(J_{\Theta\Theta}, J_{\Theta\Lambda}\xi)$}
    $R_S = R_\Lambda - J_{\Lambda\Theta}$ **gmres**$(J_{\Theta\Theta}, R_\Theta)$
    $\delta\Lambda =$ **gmres**(J_S, R_S)
    $\delta\Theta =$ **gmres**$(J_{\Theta\Theta}, R_\Theta + J_{\Theta\Lambda}\delta\Lambda)$
    $[\Theta^{n,k+1}, \Lambda^{n,k+1}] = [\Theta^{n,k} + \delta\Theta, \Lambda^{n,k} + \delta\Lambda]$
  end for
  $[\Theta^n, \Lambda^n] = [\Theta^{n,k}, \Lambda^{n,k}]$
end for

the strict interior of each subdomain (Figure 3, left), we follow the standard choice of picking the phase mobility from the upwind direction, as given in (5.1a)–(5.1b). However, at subdomain interfaces (Figure 3, right) several possibilities exist (described below), which can have a significant impact on mass conservation and nonlinear convergence of the method. For simplicity, the figure and quadrature formulas assume matching grids, but a simple modification allows the general case. Here $\boldsymbol{v}$ represents the common velocity basis function on an internal edge, and $\boldsymbol{v}^L$ and $\boldsymbol{v}^R$ represent the distinct velocity basis functions on opposing sides of an interface. We advocate usage of block-to-block upwinding, which is shown to be the most stable and accurate choice in the numerical results for the GJ method (see section 6.2).

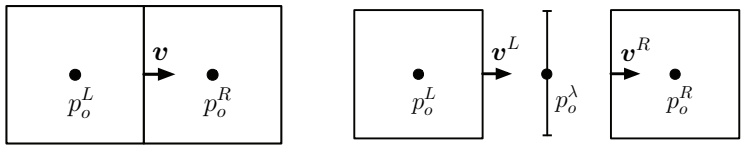

FIG. 3. *Degrees of freedom within subdomain (left) and on interface (right) for upwinding.*

Upwinding "within a subdomain." Without loss of generality, consider two neighboring elements (i, j, k) and $(i + 1, j, k)$ in the interior of a subdomain, denoted E^L and E^R, respectively. Let $\triangle p_o$ denote the difference in oil pressure of these elements, that is,

$$\triangle p_o = p_o^R - p_o^L.$$

The standard choice of upwind mobility in combination with the trapezoidal midpoint (TM) quadrature rule [5] gives the following expression for the numerical evaluation of oil mobility terms in (4.2a):

$$m_o^{up} = \begin{cases} m_o^L & \text{if } \triangle p_o < 0, \\ m_o^R & \text{if } \triangle p_o > 0, \end{cases} \tag{5.1a}$$

$$\int_{E^L \cup E^R} m_o \, \boldsymbol{v} \cdot \boldsymbol{v} dx \underset{TM}{\approx} m_o^{up} \times \left(\frac{h_x^L}{2\, h_y\, h_z} + \frac{h_x^R}{2\, h_y\, h_z} \right). \tag{5.1b}$$

Similar expressions are used for integrating water mobility terms in (4.2e). Here h_x, h_y, h_z are the mesh widths in the three coordinate directions on a brick element. However, for elements adjacent to a subdomain interface, the choice for upwinding is not obvious. In the following, we discuss three possibilities (a)–(c).

We define the L^2-projection from the traces of left and right subdomains to the mortar grid as $\mathcal{Q}_{L\Lambda}$ and $\mathcal{Q}_{R\Lambda}$, respectively, and define the L^2-projection directly from right to left subdomains as $\mathcal{Q}_{RL}$:

(5.2)
$$\mathcal{Q}_{L\Lambda} : W_h(\Omega^L|_\Gamma) \to M_H, \quad \mathcal{Q}_{R\Lambda} : W_h(\Omega^R|_\Gamma) \to M_H, \quad \mathcal{Q}_{RL} : W_h(\Omega^R|_\Gamma) \to W_h(\Omega^L|_\Gamma)$$

The inverse maps are obtained using the transpose. Different upwinding methods are obtained by using the following definitions:

(a) *Upwinding "using mortar."* This method uses the mortar unknown itself to define the upwind direction. Recall in this paper that we are using $\lambda_1 = p_o^\Gamma$ and $\lambda_2 = p_w^\Gamma$. The pressure differences are

$$\triangle p_o^L = \mathcal{Q}_{L\Lambda}^T p_o^\Gamma - p_o^L, \tag{5.3a}$$

$$\triangle p_o^R = p_o^R - \mathcal{Q}_{R\Lambda}^T p_o^\Gamma. \tag{5.3b}$$

The upwinded mobility comes from the properties on the current subdomain or a projection of the mortar variables themselves, i.e., on the left side of the interface

$$m_o^{up,L} = \begin{cases} \dfrac{k_{ro}(s_o^L)\rho_o(p_o^L)}{\mu_o} & \text{if } \triangle p_o^L < 0, \\ \dfrac{k_{ro}(\mathcal{Q}_{L\Lambda}^T s_o^\Gamma)\rho_o(\mathcal{Q}_{L\Lambda}^T p_o^\Gamma)}{\mu_o} & \text{if } \triangle p_o^L > 0, \end{cases} \tag{5.3c}$$

and on the right side of the interface

$$m_o^{up,R} = \begin{cases} \dfrac{k_{ro}(\mathcal{Q}_{R\Lambda}^T s_o^\Gamma)\rho_o(\mathcal{Q}_{R\Lambda}^T p_o^\Gamma)}{\mu_o} & \text{if } \triangle p_o^R < 0, \\ \dfrac{k_{ro}(s_o^R)\rho_o(p_o^R)}{\mu_o} & \text{if } \triangle p_o^R > 0. \end{cases} \tag{5.3d}$$

(b) *Upwinding "through mortar."* This method uses the trace of the adjacent subdomain unknown projected through the mortar space onto the current subdomain grid to define the upwind direction. The pressure differences are

$$\triangle p_o^L = \mathcal{Q}_{L\Lambda}^T \mathcal{Q}_{R\Lambda} p_o^R - p_o^L, \tag{5.4a}$$

$$\triangle p_o^R = p_o^R - \mathcal{Q}_{R\Lambda}^T \mathcal{Q}_{L\Lambda} p_o^L. \tag{5.4b}$$

The upwinded mobility comes from the properties on the current subdomain or a projection of the values of the adjacent subdomain through the mortar space onto the current subdomain, i.e., on the left side of the interface

$$m_o^{up,L} = \begin{cases} \dfrac{k_{ro}(s_o^L)\rho_o(p_o^L)}{\mu_o} & \text{if } \triangle p_o^L < 0, \\ \dfrac{k_{ro}(\mathcal{Q}_{L\Lambda}^T \mathcal{Q}_{R\Lambda} s_o^R)\rho_o(\mathcal{Q}_{L\Lambda}^T \mathcal{Q}_{R\Lambda} p_o^R)}{\mu_o} & \text{if } \triangle p_o^L > 0, \end{cases} \tag{5.4c}$$

and on the right side of the interface

$$m_o^{up,R} = \begin{cases} \dfrac{k_{ro}(\mathcal{Q}_{R\Lambda}^T \mathcal{Q}_{L\Lambda} s_o^L)\rho_o(\mathcal{Q}_{R\Lambda}^T \mathcal{Q}_{L\Lambda} p_o^L)}{\mu_o} & \text{if } \triangle p_o^R < 0, \\ \dfrac{k_{ro}(s_o^R)\rho_o(p_o^R)}{\mu_o} & \text{if } \triangle p_o^R > 0. \end{cases} \tag{5.4d}$$

(c) *Upwinding "block to block."* This method uses the trace of the adjacent subdomain unknown projected directly onto the current subdomain to define the upwind direction. The pressure differences are

$$\triangle p_o^L = \mathcal{Q}_{RL} p_o^R - p_o^L, \tag{5.5a}$$

$$\triangle p_o^R = p_o^R - \mathcal{Q}_{RL}^T p_o^L \tag{5.5b}$$

The upwinded mobility comes from the properties on the current subdomain or a projection of the values of the adjacent subdomain directly onto the current subdomain, i.e., on the left side of the interface

$$m_o^{up,L} = \begin{cases} \dfrac{k_{ro}(s_o^L)\rho_o(p_o^L)}{\mu_o} & \text{if } \triangle p_o^L < 0, \\ \dfrac{k_{ro}(\mathcal{Q}_{RL} s_o^R)\rho_o(\mathcal{Q}_{RL} p_o^R)}{\mu_o} & \text{if } \triangle p_o^L > 0, \end{cases} \tag{5.5c}$$

and on the right side of the interface

$$m_o^{up,R} = \begin{cases} \dfrac{k_{ro}(\mathcal{Q}_{RL}^T s_o^L)\rho_o(\mathcal{Q}_{RL}^T p_o^L)}{\mu_o} & \text{if } \triangle p_o^R < 0, \\ \dfrac{k_{ro}(s_o^R)\rho_o(p_o^R)}{\mu_o} & \text{if } \triangle p_o^R > 0. \end{cases} \tag{5.5d}$$

Using the definitions in upwinding methods (a)–(c), the application of trapezoidal midpoint quadrature in the elements adjacent to subdomain interfaces gives

$$\int_{E^L} m_o \boldsymbol{v}^L \cdot \boldsymbol{v}^L dx \underset{TM}{\approx} m_o^{up,L} \times \left(\frac{h_x^L}{2\, h_y\, h_z} \right),$$

$$\int_{E^R} m_o \boldsymbol{v}^R \cdot \boldsymbol{v}^R dx \underset{TM}{\approx} m_o^{up,R} \times \left(\frac{h_x^R}{2\, h_y\, h_z} \right)$$

for the subdomains to the left and to the right of the interface, respectively.

Note that in the case of matching grids, all three projections in (5.2) are identity. The consequence is that upwinding methods (b) and (c) reduce to the single domain case and produce the same solution for fluid transport. Conversely, in method (a), the definitions of the upwind direction and mobility to the left and right of the interface are not equal. Thus even in the case of matching grids, the solution may deviate from a comparable single-domain method, the nonlinear iteration may diverge, or the linear systems may become singular. In the authors' experience, the difficulties with method (a) are aggravated in the case of nonmatching grids (which is one of the main benefits of using domain decomposition), and also as the number of subdomain interfaces is increased for a fixed problem size.

The reason we advocate method (c) over method (b) is that the mortar grid size H is usually taken to be coarser than the subdomain grid size h, giving the multiscale

aspect to the method. Therefore, the two subsequent projections in method (b) may cause an unnecessary loss of information in the fluid properties when needed from the adjacent subdomain. This can be entirely prevented using method (c), where the fluid properties use the best possible information.

One other important remark is that mortar saturation values are needed in method (a). To get mortar saturation values when using two phase pressures as mortar unknowns, this requires an elementwise Newton iteration to invert the p_c curve in the elements along the interface. This also means that the case $p_c = 0$ cannot be handled. Since methods (b) and (c) do not require mortar saturation values, the additional Newton iteration is no longer necessary, the zero capillary pressure case can be handled, and two interface pressures can be used as mortar unknowns (which is the most natural case when integrating by parts in the variational form).

6. Numerical examples. In this section we perform two numerical examples with the multiscale mortar method. Example 1 (section 6.1) shows the capability of modeling spatially nonconforming domains and strong heterogeneities and compares the efficiency of the GJ algorithm to the FD algorithm (recall Figure 2). Example 2 (section 6.2) shows the capability of modeling multiple rock types with strong capillarity and compares the accuracy of water saturation solutions using the various upwinding methods. In both examples, the fluid viscosities are $\mu_o = 2.0$ cp and $\mu_w = 0.5$ cp, the fluid compressibilities are $c_o = 4.0\text{e-}3$ psi^{-1} and $c_w = 3.3\text{e-}4$ psi^{-1}, and the fluid reference densities are $\rho_o^{\text{ref}} = 56.0$ lb/ft^3 and $\rho_w^{\text{ref}} = 62.3$ lb/ft^3.

6.1. Example 1: Fault. This example has nonconforming spatial geometry in the form of a fault between two subdomains with different mesh resolutions. This highlights the ability of the multiscale mortar method to resolve complex geologic features such as faults and lithology. The geometry and discretization are shown in Figure 4, with positive x representing the depth direction. The size of subdomain Ω^1 is $20 \times 100 \times 100$ ft with a uniform grid of $10 \times 10 \times 10$ elements. The subdomain Ω^2 is offset by 8 ft in the x-direction and 10 ft in the z-direction and has a size of $21 \times 100 \times 105$ ft with a uniform grid of $7 \times 7 \times 7$ elements. The mortar grid is defined in the area of overlap on the fault plane with a uniform grid of 2×5 elements and uses a continuous linear mortar.

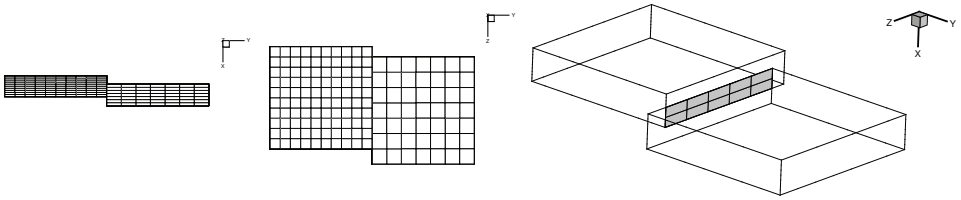

FIG. 4. *Geometry and discretization for Example* 1: *subdomain grids in the x-y plane (left), subdomain grids in y-z plane (middle), and mortar grid (right).*

On this grid we have two absolute permeability datasets for the x-component of the diagonal tensor K, shown in Figure 5. The y- and z-components of this tensor multiply the x-component by a factor of 2.

The first dataset is denoted the barrier case with a background permeability of 100 md, a high permeability streak of 1000 md, and four low permeability barriers of 0.1 md. On subdomain Ω^1, the streak is on x-layer 6, and the barriers are on

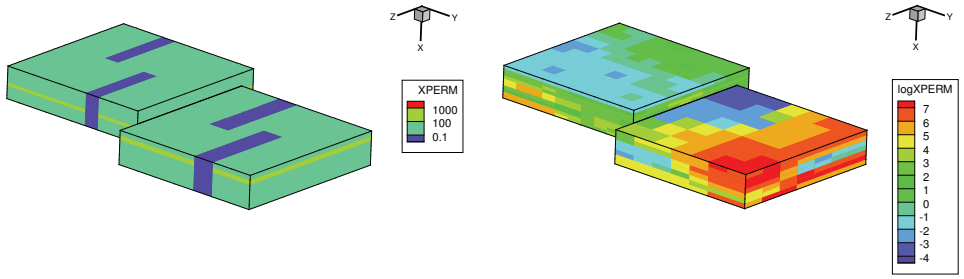

FIG. 5. *Absolute permeability datasets for Example* 1*: Barrier case (left) and logarithm of heterogeneous case (right).*

y-columns 4 and 7. On subdomain Ω^2, the streak is on x-layer 2, and the barriers are on y-columns 3 and 5. The porosity in this case is $\phi = 0.2$.

The second dataset is denoted the heterogeneous case where both the permeability and porosity values are used from a subset of the SPE10 industrial benchmark [9]. On subdomain Ω^1, the values span from data locations $(21, 1, 1)$ to $(30, 10, 10)$. On subdomain Ω^2, the values span from data locations $(61, 1, 1)$ to $(67, 7, 7)$. Recall that the x-coordinate represents depth.

The residual saturations for oil and water are $s_{rw} = 0.2$ and $s_{ro} = 0.2$, respectively. The effective saturation $s_e \in (0, 1)$ is defined as

$$s_e = \frac{s_w - s_{rw}}{1 - s_{rw} - s_{ro}}. \tag{6.1}$$

There is one injection well and one production well in opposite corners of the domain using a Peaceman well model [8]. Both wells are vertical and completed through the entire reservoir depth. The injection bottom hole pressure varies linearly from (505 psi, day 0) to (510 psi, day 30) and remains constant thereafter. The production bottom hole pressure varies linearly from (495 psi, day 0) to (490 psi, day 30) and remains constant thereafter. External boundary conditions are no-flow. The initial condition comes from a hydrostatic equilibrium with $p_o(0) = 500$ psi and $s_w(0) = 0.22$. There is a uniform time step of $\delta t = 1$ day with a final simulation time of $T = 400$ days. The capillary pressure, relative oil permeability, and relative water permeability follow the following J-Leverett and Brooks–Corey relationships [8] with entry pressure $p_d = 10$ md and grain size exponent $\lambda = 2$:

$$p_c(s_w) = p_d\, s_e^{-1/\lambda}, \quad k_{ro}(s_w) = (1 - s_e)^2(1 - s_e^{(2+\lambda)/\lambda}), \quad k_{rw}(s_w) = s_e^{(2+3\lambda)/\lambda}. \tag{6.2}$$

Plots of the simulation results are shown in Figure 6. Many effects contribute to the observed complex behavior, including the presence of the fault, gravity, capillarity, compressibility, anisotropy, and heterogeneity. In the barrier case, the water saturation front must snake around the low permeability barriers to travel from the injection to production well. In the heterogeneous case, the water saturation contour iso-surfaces are rougher from increased heterogeneity and travel slower due to lower permeability values.

We demonstrate the efficiency of the global Jacobian algorithm proposed in this work for two-phase flow with a comparison of the global Jacobian (GJ) method with the previously developed forward difference (FD) method. Since the algorithms are

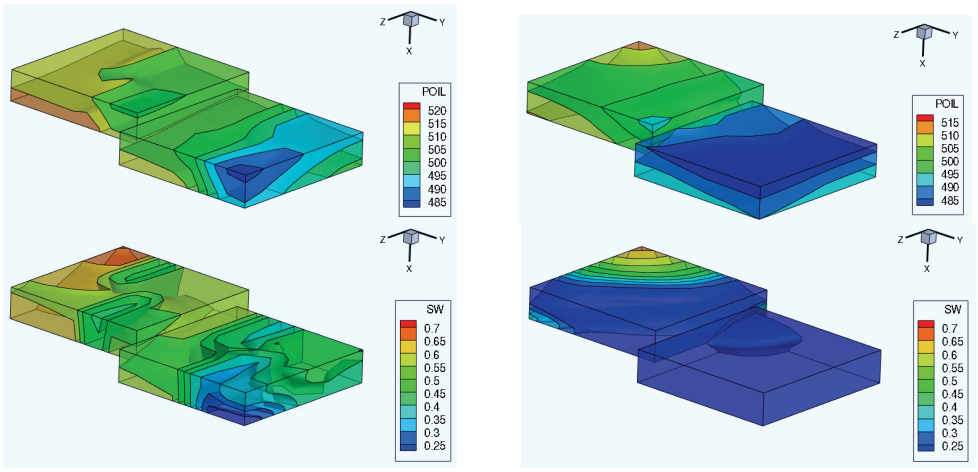

FIG. 6. *Numerical result for Example* 1 *at* $T = 400$ *days: Top row shows oil pressure and bottom row shows water saturation, with a transparent domain and iso-surfaces shown at contour levels. Left column shows barrier case and right column shows heterogeneous case.*

TABLE 1
Relative tolerances for the algorithms used in Example 1.

FD method		GJ method	
Forward difference	1E-7	Global Newton	1E-4
Interface Newton	1E-4	Direct linear solver	
Interface GMRES	1E-2		
Subdom. Newton	1E-3		
Subdom. GMRES	1E-4		

inherently different in the iterations that are performed, in Table 1 we summarize the tolerances used for each algorithm. In each method, the outermost Newton tolerance is taken to be the same, which produces solutions with comparable accuracy. The FD method requires choosing five tolerances in order to strike a careful balance between achieving convergence without oversolving inner iterations. In practice, this requires some trial and error, and tolerances are shown that worked well for this particular problem. Conversely, the GJ method is conceptually simpler as it only needs a nonlinear and a linear tolerance, akin to single domain methods. In practice, this also makes it much more robust. In this work, we perform a single-processor simulation and use a sparse direct solver for the linear systems in the GJ method. Numerical results were not yet obtained with the GJS method due to the use of the direct solver. We note that the code is written in parallel following the previous work on slightly compressible single-phase flow [14], but that a parallel preconditioner appropriate for a two-phase flow system [36, 25] was not yet available in the implementation.

The computational cost for these two algorithms and two cases are summarized in Table 2. The simulation results show that the number of interface Newton iterations with the FD method is comparable to the number of global Newton iterations with the GJ method, in both the barrier and heterogeneous cases. However, the FD method requires a large number of inner iterations that are not present in the GJ method (recall Figure 2). For example, each time a subdomain Newton iteration is performed

TABLE 2

Computational cost for Example 1 *measured in terms of iteration counts and CPU time until* $t = 200$ *days. Tot. is the total iteration count for the entire simulation, Avg.* 1 *is the average number of iterations per time step, and Avg.* 2 *is the average number of iterations per next outermost iteration. Nonlinear subdomain solves are performed by solving a global block diagonal nonlinear system with blocks corresponding to subdomains, using a global Newton iteration. All subdomain computations are decoupled due to the block diagonal structure of the system, and subdomain Newton iterations are reported as the number of iterations for the global Newton solve.*

FD method									
Perm.	Intf. Newton		Intf. GMRES			Subdom. Newton			CPU
	Tot.	Avg. 1	Tot.	Avg. 1	Avg. 2	Tot.	Avg. 1	Avg. 2	time
Barrier	331	1.66	6,355	31.78	19.20	20,662	103.31	3.25	161.49
Heterog.	241	1.21	2,629	13.15	10.91	9,212	46.06	3.50	71.18

GJ method			
Perm.	Global Newton		CPU
	Tot.	Avg. 1	time
Barrier	342	1.71	11.80
Heterog.	212	1.06	7.71

for the Barrier problem, it takes an average of 3.25 iterations (Avg. 2), but since it is inside two outer loops, the average number of subdomain Newton iterations per time step is actually 103.31 (Avg. 1). The expense of these nested iterations is evident in the CPU time, where the FD method is observed to take an order of magnitude longer to run using the same system and optimization level. However, we note that since CPU times are highly subjective (they are dependent on machine, implementation, problem size, tolerances, etc.), the iteration counts provide a stronger basis to argue that the GJ method outperforms the FD method.

6.2. Example 2: Multiple rock types. This example is based on a lab experiment by Kueper in [24] with multiple rock types and strong capillary effects. Here we perform a simplified version of this experiment from [40] with two rock types. The domain is $\Omega = 70 \times 50$ ft with a fine grid resolution of 1×1 ft. This grid is divided into $N_\Omega = 2 \times 2 = 4$ subdomains with a coarse mortar such that $H = h^{2/3}$ with continuous linear elements as shown in Figure 7. There are two injection wells at the top center of the domains with a bottom hole pressure of 1600 psi, and there are a total of ten production wells spaced evenly on the left and right sides of the domain with a bottom hole pressure of 1000 psi. External boundary conditions are no-flow.

The capillary pressure curve for each rock type is given by (6.3a) as a continuous piecewise function with cutoff saturations $s_{c1} = 0.01$ and $s_{c2} = 0.9$, effective saturation s_e given by (6.1), and residual saturations equal to $s_{ro} = 0.05$ and $s_{rw} = 0.2$. The entry pressures p_d and the grain size exponents λ vary for each rock type according to (6.3b), along with absolute permeability K and porosity ϕ. Relative permeabilities are also given in (6.3b). The initial condition is taken to be uniform $p_o(0) = 500$ psi and $s_w(0) = 0.2$. The external boundary conditions are no-flow. The time step is gradually ramped up from min $\delta t = 0.001$ day to max $\delta t = 0.1$ day with a multiplier of 1.05 and a final simulation time of $T = 200$ days.

$$p_c(s_w) = \begin{cases} p_d\, s_{c1}^{-1/\lambda} & \text{if } 0 \le s_e < s_{c1}, \\ p_d\, s_e^{-1/\lambda} & \text{if } s_{c1} \le s_e \le s_{c2}, \\ p_d\, s_{c2}^{-1/\lambda}\, \frac{1-s_e}{1-s_{c2}} & \text{if } s_{c2} < s_e \le 1, \end{cases} \tag{6.3a}$$

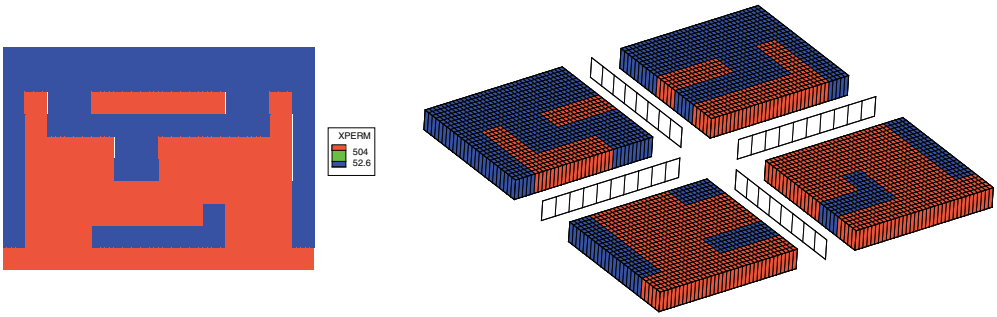

FIG. 7. *Absolute permeability (left) and an exploded view of the division into four subdomains with coarse mortars (right) for Example* 2.

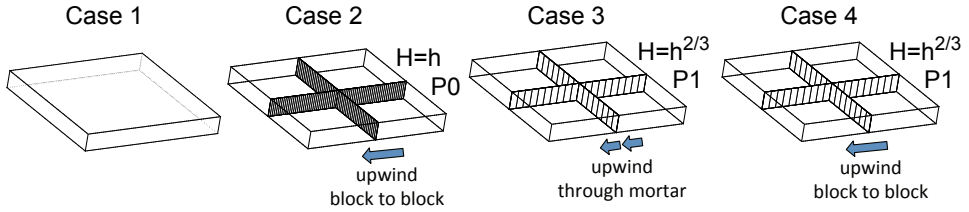

FIG. 8. *Four test cases used in Example* 2.

(6.3b)

	p_d	λ	K	ϕ
Rock type 1	135 psi	2.49	504 md	0.2
Rock type 2	37.7 psi	3.86	52.6 md	0.2

$$k_{rw} = 0.9\, s_e^2$$
$$k_{ro} = 0.5\,(1 - s_e)^2$$

We run this example with four test cases, illustrated in Figure 8. Case 1 is a single domain used as a reference solution and verification with [40]. Cases 2–4 all use four subdomains with the GJ method. Case 2 uses a mortar space on a matching fine scale $H = h$ with piecewise constant elements and employs upwinding "block to block." Cases 3 and 4 both use a mortar space on a coarse scale $H = h^{2/3}$ with continuous linear elements. Case 3 employs upwinding "through mortar," and Case 4 employs upwinding "block to block." There is no case for upwinding "using mortar" because it could not achieve nonlinear convergence. The Newton tolerance is 1e-2.

Oil pressure and water saturation fields are shown in Figure 9 at $t = 1, 100, 200$ days for Case 2. These fields are representative of the results in [40], with additional effects due to fluid compressibility. They are observed to be highly discontinuous at rock-type boundaries due to the strong capillarity effects that present in the physical model. Additional numerical discontinuities appear in the solution on subdomain interfaces, which is a consequence of mass conservation enforced weakly in the mortar space.

Errors in water saturation at the final time are shown in Figure 10. The error between the reference solution Case 1 and the fine-scale mortar solution Case 2 shows an excellent match with a pointwise error of $\|s_w^2 - s_w^1\|_\infty = 0.001$, which is below nonlinear solver tolerance. This uses block-to-block upwinding and was the best result that was obtained with four subdomains. Case 2 is then compared to the two coarse mortar cases when varying the upwinding technique. The comparison against the coarse mortar using upwinding through a mortar shows a very large pointwise error of $\|s_w^3 - s_w^2\|_\infty = 0.37$. The figure shows error accumulating near injection wells,

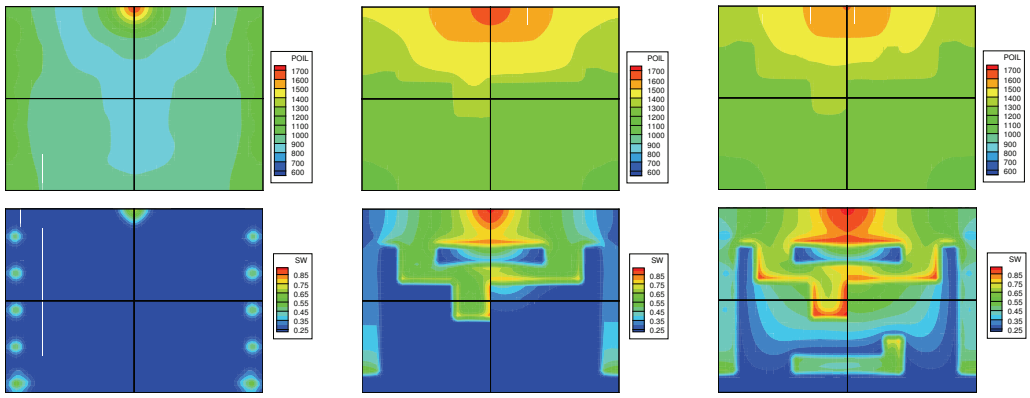

FIG. 9. *Oil pressure (top row) and water saturation (bottom row) for Example* 2 *with Case* 2. *Times shown for* $t = 1$ *days (left),* $t = 100$ *days (center), and* $t = 200$ *days (right).*

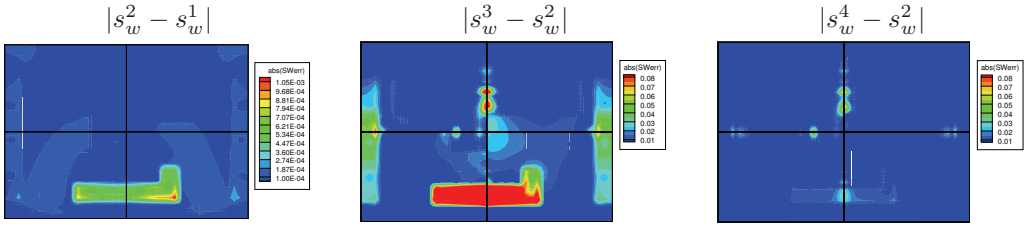

FIG. 10. *Errors in water saturation at* $t = 200$ *days for Example* 2: *Case* 2 *versus* 1 *(left), Case* 3 *versus* 2 *(center), and Case* 4 *versus* 2 *(right).*

rock-type boundaries, and well within the subdomains. No comparison could be shown with upwinding using mortar itself because nonlinear convergence could not be achieved, giving evidence that this is not the proper choice. The comparison against the coarse mortar using the block-to-block upwinding technique shows a significant improvement with a pointwise error of $\|s_w^4 - s_w^2\|_\infty = 0.07$. Moreover, the figure shows errors have completely disappeared around the injection wells and in the interior of the domain, with just some modest error remaining on the interfaces. This gives strong evidence that block-to-block upwinding provides improved transport behavior in the nonoverlapping domain decomposition setting.

7. Conclusions. We have developed several new algorithms for efficient and robust solution to the multiscale mortar mixed finite element method in a two-phase fully implicit formulation with capillarity, gravity, and compressibility. The novelty of our approach is to linearize the coupled system of subdomain and mortar variables simultaneously to form a global Jacobian. The flux unknowns (and optionally subdomain unknowns) are eliminated to obtain linear systems for the 1st (or the 2nd) Schur complements, denoted by the GJ and GJS methods, respectively. The 1st Schur complement system has a modular block structure comprising previously available subdomain Jacobian matrices with sparse coupling blocks. The 2nd Schur complement system can utilize an extensive collection of previously developed preconditioners for a linear interface problem. In addition we have investigated multiple techniques for accurate integration of phase mobility terms near subdomain interfaces and have found the best method to be block-to-block upwinding. Numerical results demonstrate the modeling flexibility of the mortar method, the efficiency of the GJ

algorithm, and an improvement to fluid saturation accuracy with the proposed upwinding technique. These example include important modeling features such as nonmatching grids, geologic faults, permeability barriers, strong heterogeneities, multiple rock types, and three-dimensional domains. Future plans include developing specialized preconditioners for the GJ and GJS methods and extensions to more complex physical models and discretizations.

Acknowledgment. We would like to thank Mika Juntunen for his instrumental help in formulating the original GJ concept.

REFERENCES

[1] J. E. Aarnes, S. Krogstad, and K.-A. Lie, *A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids*, Multiscale Model. Simul., 5 (2006), pp. 337–363.

[2] T. Arbogast and S. L. Bryant, *Numerical subgrid upscaling for waterflood simulations*, in SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2001.

[3] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, *Mixed finite element methods on nonmatching multiblock grids*, SIAM J. Numer. Anal., 37 (2000), pp. 1295–1315.

[4] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, *A multiscale mortar mixed finite element method*, Multiscale Model. Simul., 6 (2007), pp. 319–346.

[5] T. Arbogast, M. F. Wheeler, and I. Yotov, *Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences*, SIAM J. Numer. Anal., 34 (1997), pp. 828–852.

[6] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[7] F. Brunner, F. A. Radu, and P. Knabner, *Analysis of an upwind-mixed hybrid finite element method for transport problems*, SIAM J. Numer. Anal., 52 (2014), pp. 83–102.

[8] Z. Chen, G. Huan, and Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, Comput. Sci. Eng. 2, SIAM, Philadelphia, 2006.

[9] M. A. Christie and M. J. Blunt, *Tenth SPE comparative solution project: A comparison of upscaling techniques*, SPE Reservoir Eval. Engrg., 4 (2001), pp. 308–317.

[10] C. N. Dawson, H. Klíe, M. F. Wheeler, and C. S. Woodward, *A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver*, Comput. Geosci., 1 (1997), pp. 215–249 (1998).

[11] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing, *Accurate multiscale finite element methods for two-phase flow simulations*, J. Comput. Phys., 220 (2006), pp. 155–174.

[12] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[13] J. Galvis and Y. Efendiev, *Domain decomposition preconditioners for multiscale flows in high-contrast media*, Multiscale Model. Simul., 8 (2010), pp. 1461–1483.

[14] B. Ganis, M. Juntunen, G. Pencheva, M. F. Wheeler, and I. Yotov, *A global Jacobian method for mortar discretizations of nonlinear porous media flows*, SIAM J. Sci. Comput., 36 (2014), pp. A522–A542.

[15] B. Ganis, G. Pencheva, M. F. Wheeler, T. Wildey, and I. Yotov, *A frozen Jacobian multiscale mortar preconditioner for nonlinear interface operators*, Multiscale Model. Simul., 10 (2012), pp. 853–873.

[16] B. Ganis and I. Yotov, *Implementation of a mortar mixed finite element method using a multiscale flux basis*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 3989–3998.

[17] V. Girault, G. Pencheva, M. F. Wheeler, and T. Wildey, *Domain decomposition for poroelasticity and elasticity with DG jumps and mortars*, Math. Models Methods Appl. Sci., 21 (2011), pp. 169–213.

[18] V. Girault, S. Sun, M. F. Wheeler, and I. Yotov, *Coupling discontinuous Galerkin and mixed finite element discretizations using mortar finite elements*, SIAM J. Numer. Anal., 46 (2008), pp. 949–979.

[19] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comput. Phys., 134 (1997), pp. 169–189.

[20] P. Jenny, S. H. Lee, and H. A. Tchelepi, *Multi-scale finite-volume method for elliptic problems in subsurface flow simulation*, J. Comput. Phys., 187 (2003), pp. 47–67.

[21] L. Jiang, J. E. Aarnes, and Y. Efendiev, *Some multiscale results using limited global infor-*

mation for two-phase flow simulations, Int. J. Numer. Anal. Model., 9 (2012), pp. 115–131.
[22] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995.
[23] A. Klawonn, M. Lanser, P. Radtke, and O. Rheinbach, *On an adaptive coarse space and on nonlinear domain decomposition*, in Proceedings on 21st International Conference on Domain Decomposition Methods, INRIA Rennes-Bretagne-Atlantique, 2012.
[24] B. H. Kueper, W. Abbott, and G. Farquhar, *Experimental observations of multiphase flow in heterogeneous porous media*, J. Contaminant Hydrology, 5 (1989), pp. 83–95.
[25] S. Lacroix, Yu. Vassilevski, J. Wheeler, and M. Wheeler, *Iterative solution methods for modeling multiphase flow in porous media fully implicitly*, SIAM J. Sci. Comput., 25 (2003), pp. 905–926.
[26] K. Lipnikov, J. D. Moulton, and D. Svyatskiy, *Adaptive strategies in the multilevel multiscale mimetic (M^3) method for two-phase flows in porous media*, Multiscale Model. Simul., 9 (2011), pp. 991–1016.
[27] Q. Lu, M. Peszyńska, and M. F. Wheeler, *A parallel multiblock black-oil model in multimodel implementation*, SPE J., 7 (2002), pp. 278–287.
[28] Y. Mehmani and M. T. Balhoff, *Bridging from pore to continuum: A hybrid mortar domain decomposition framework for subsurface flow and transport*, Multiscale Model. Simul., 12 (2014), pp. 667–693.
[29] J.-C. Nédélec, *Mixed finite elements in R^3*, Numer. Math., 35 (1980), pp. 315–341.
[30] G. Pencheva and I. Yotov, *Balancing domain decomposition for mortar mixed finite element methods*, Numer. Linear Algebra Appl., 10 (2003), pp. 159–180.
[31] M. Peszyńska, M. F. Wheeler, and I. Yotov, *Mortar upscaling for multiphase flow in porous media*, Comput. Geosci., 6 (2002), pp. 73–100.
[32] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Numer. Math. Sci. Comput. 10, Clarendon Press, Oxford, 1999.
[33] R. A. Raviart and J. M. Thomas, *A mixed finite element method for 2nd order elliptic problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Math. 606, Springer-Verlag, New York, 1977, pp. 292–315.
[34] A. Toselli and O. B. Widlund, *Domain Decomposition Methods: Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer-Verlag, Berlin, 2005.
[35] D. Vassilev and I. Yotov, *Coupling Stokes–Darcy flow with transport*, SIAM J. Sci. Comput., 31 (2009), pp. 3661–3684.
[36] J. R. Wallis, R. P. Kendall, and T. E. Little, *Constrained residual acceleration of conjugate residual methods*, in SPE Reservoir Simulation Symposium, 1985, SPE 13536.
[37] J. A. Wheeler, M. F. Wheeler, and I. Yotov, *Enhanced velocity mixed finite element methods for flow in multiblock domains*, Comput. Geosci., 6 (2002), pp. 315–332.
[38] M. F. Wheeler, T. Arbogast, S. Bryant, J. Eaton, Q. Lu, M. Peszynska, and I. Yotov, *A parallel multiblock/multidomain approach for reservoir simulation*, in SPE Reservoir Simulation Symposium, 1999, SPE 51884-MS.
[39] M. F. Wheeler, G. Pencheva, and S. G. Thomas, *Mortar coupling of multiphase flow and reactive transport on non-matching grids*, in Proceedings of the 5th International Symposium on Finite Volumes for Complex Applications, 2008, pp. 135–143.
[40] M. F. Wheeler, G. Xue, and I. Yotov, *Accurate cell-centered discretizations for modeling multiphase flow in porous media on general hexahedral and simplicial grids*, SPE J., 17 (2012), pp. 779–793, SPE 141534-PA.
[41] M. F. Wheeler, G. Xue, and I. Yotov, *A multiscale mortar multipoint flux mixed finite element method*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 759–796.
[42] M. F. Wheeler and I. Yotov, *Multigrid on the interface for mortar mixed finite element methods for elliptic problems*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 287–302.
[43] B. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, Lect. Notes Comput. Sci. Eng. 17, Springer-Verlag, Berlin, 2001.
[44] M. Wolff, B. Flemisch, and R. Helmig, *An adaptive multiscale approach for modeling two-phase flow in porous media including capillary pressure*, Water Resources Res., 49 (2013), pp. 8139–8159.
[45] I. Yotov, *A multilevel Newton–Krylov interface solver for multiphysics couplings of flow in porous media*, Numer. Linear Algebra Appl., 8 (2001), pp. 551–570.
[46] I. Yotov, *Interface solvers and preconditioners of domain decomposition type for multiphase flow in multiblock porous media*, in Scientific Computing and Applications, Adv. Comput. Theory Pract. 7, Nova Sci. Publ., Huntington, NY, 2001, pp. 157–167.