A SCALABLE MULTIGRID REDUCTION FRAMEWORK FOR MULTIPHASE POROMECHANICS OF HETEROGENEOUS MEDIA[*]

QUAN M. BUI[†], DANIEL OSEI-KUFFUOR[‡], NICOLA CASTELLETTO[§], AND JOSHUA A. WHITE[§]

Abstract. Simulation of multiphase poromechanics involves solving a multiphysics problem in which multiphase flow and transport are tightly coupled with the porous medium deformation. To capture this dynamic interplay, fully implicit methods, also known as monolithic approaches, are usually preferred. The main bottleneck of a monolithic approach is that it requires solution of large linear systems that result from the discretization and linearization of the governing balance equations. Because such systems are nonsymmetric, indefinite, and highly ill-conditioned, preconditioning is critical for fast convergence. Recently, most efforts in designing efficient preconditioners for multiphase poromechanics have been dominated by physics-based strategies. Current state-of-the-art "black-box" solvers such as algebraic multigrid (AMG) are ineffective because they cannot effectively capture the strong coupling between the mechanics and the flow subproblems, as well as the coupling inherent in the multiphase flow and transport process. In this work, we develop an algebraic framework based on multigrid reduction (MGR) that is suited for tightly coupled systems of PDEs. Using this framework, the decoupling between the equations is done algebraically through defining appropriate interpolation and restriction operators. One can then employ existing solvers for each of the decoupled blocks or design a new solver based on knowledge of the physics. We demonstrate the applicability of our framework when used as a "black-box" solver for multiphase poromechanics. We show that the framework is flexible to accommodate a wide range of scenarios, as well as efficient and scalable for large problems.

Key words. multigrid reduction, algebraic multigrid, multiphase flow, poromechanics, porous media

AMS subject classifications. 65M55, 76S05, 74F10, 76T99

DOI. 10.1137/19M1256117

1. Introduction. Modeling subsurface systems requires an understanding of many different physical processes, including multiphase fluid flow and transport, and geomechanical deformations. These processes are often tightly coupled in a "two-way" fashion: for example, the flow process can have a large influence on the mechanical process and in turn can be affected by the feedback of the produced mechanical response. To simulate these processes, one needs to solve a set of coupled, nonlinear, time-dependent PDEs that govern the conservation of mass of the fluid phases and linear momentum of the solid-fluid mixture. For this system, fully implicit time dis-

cretization is the widely preferred approach because it is unconditionally stable and allows for large time steps. However, using an implicit approach, one must solve a large, sparse, and ill-conditioned linear system at each nonlinear iteration. Robust and scalable solvers are therefore needed for large-scale simulations on high performance computing platforms. This paper presents our efforts to design an efficient preconditioning strategy based on an algebraic framework that is flexible and capable of addressing the inherent ill-conditioning of a complicated multiphysics system.

In recent years, much of the work in developing preconditioning strategies for coupled problems has focused on so-called physics-based strategies. The key is to use knowledge of the specific physical processes involved to break the tightly coupled systems into smaller subproblems whose properties are well studied. For example, these subproblems could take the form of an elliptic, hyperbolic, or parabolic PDE, to which appropriate techniques can be applied. For fully implicit simulation of complex multiphase flow and transport without mechanics, one of the most popular methods is the *constrained residual pressure* (CPR) multistage preconditioning technique [45, 46]. For single-phase flow poromechanics, many block preconditioners have been developed [1, 5, 6, 24, 30, 48, 50]. In the context of multiphase poromechanics, one recent strategy [49] uses the *fixed-stress* splitting [27, 33, 40, 50] of the mechanics and the flow parts combined with a CPR approach [45] for the multiphase flow system. In general, these physics-based preconditioners are among the most effective techniques available. However, designing a good strategy is both time-consuming and challenging as it requires extensive knowledge of the particular continuous model of interest. One alternative is to use a "black-box" approach, such as *algebraic multigrid* (AMG) [42, 44]. AMG techniques are among the most efficient and scalable methods for solving sparse linear systems. Unlike *geometric multigrid*, these methods do not need an explicit hierarchy of computational grids. However, they are originally designed for scalar elliptic PDEs, and their applicability is much more limited for strongly coupled systems of PDEs.

Recently, *multigrid reduction* (MGR) [36, 37], a variant of AMG, has been applied successfully to coupled systems of multiphase flow and transport with phase transitions [8, 47]. Drawing on this success, in this work we further develop MGR into a general multilevel framework for solving discrete systems coming from discretization of tightly coupled PDEs. We also introduce a new dropping strategy for computing the reduction onto the coarse grid within the MGR V-cycle that effectively captures the coupling between mechanics and flow. The goal of this strategy is two-pronged: (1) to keep the coarse grid sparse during multilevel reduction, and (2) to make the coarse grid amenable to classical AMG. We show that with this new feature, MGR is effective as a general-purpose algebraic solver for multiphase poromechanics, and it also scales well with problem size. The rest of the paper is organized as follows. Sections 2 and 3 introduce the governing equations of multiphase poromechanics and the discretization scheme. Section 4 describes the nonlinear solution algorithm. In section 5, we describe the MGR framework and how it is applied to solve the linear systems coming from the linearization. We show numerical results in section 6 to demonstrate the robustness and scalability of the proposed preconditioner. We then end with some concluding remarks and directions for future work.

2. Problem statement. We focus on a displacement-saturation-pressure formulation of immiscible two-phase flow through a deforming poroelastic medium [14]. We limit the discussion to quasi-static small-strain kinematics. Let the subscripts w and nw denote the wetting and nonwetting fluid phase, respectively. Since the

medium's pore space is always fluid-filled, the fluid phase saturations must always sum to unity, i.e., $(s_w + s_{nw}) = 1$. Here, the wetting fluid phase saturation, denoted from now on by lower case s without subscript, is used as a primary unknown. Capillary pressure, which is the difference between the phase pressure of the non-wetting phase and the wetting phase, is not considered—a frequent assumption in many practical engineering applications. Hence, we have $p_w = p_{nw} = p$.

For a given closed domain $\overline{\Omega} = \Omega \cup \Gamma \in \mathbb{R}^3$, with Ω an open set and Γ its boundary, and time interval $\mathbb{I} = (0, t_{\max}]$, the strong form of the multiphase poromechanical initial/boundary value problem (IBVP) consists of finding the displacement vector field $\boldsymbol{u} : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}^3$, the wetting fluid phase saturation $s : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}^3$, and the pore pressure $p : \overline{\Omega} \times \mathbb{I} \to \mathbb{R}$ such that [14]

$$
\begin{aligned}
&\text{(2.1a)} \quad -\nabla \cdot \boldsymbol{\sigma} = \rho\boldsymbol{g} && \text{on } \Omega \times \mathbb{I} \quad \text{(linear momentum balance)},\\
&\text{(2.1b)} \quad \dot{m}_w + \nabla \cdot \boldsymbol{w}_w = q_w && \text{on } \Omega \times \mathbb{I} \quad \text{(wetting fluid phase mass balance)},\\
&\text{(2.1c)} \quad \dot{m}_{nw} + \nabla \cdot \boldsymbol{w}_{nw} = q_{nw} && \text{on } \Omega \times \mathbb{I} \quad \text{(non-wetting fluid phase mass balance)},
\end{aligned}
$$

where

- $\boldsymbol{\sigma} = (\mathsf{C} : \nabla^s \boldsymbol{u} - bp\mathbf{1})$ is the total Cauchy stress tensor, with C the rank-4 elasticity tensor, b the Biot coefficient, and $\mathbf{1}$ the rank-2 identity tensor;
- $\rho\boldsymbol{g}$ is a body force due to the self-weight of the multiphase mixture, with $\rho = ((1-\phi)\rho_s + \phi\rho_w s + \phi\rho_{nw}(1-s))$ the density of the mixture, ϕ the porosity, ρ_s, ρ_w, ρ_{nw} the density of the solid, the wetting, and the non-wetting fluid phase, respectively, and $\boldsymbol{g}$ the gravity vector;
- $m_w = (\phi\rho_w s)$ and $m_{nw} = (\phi\rho_{nw}(1-s))$ denote wetting and non-wetting fluid phase mass per unit volume;
- $\boldsymbol{w}_w = -(\rho_w \lambda_w \boldsymbol{\kappa} \cdot \nabla\Phi_w)$ and $\boldsymbol{w}_{nw} = -(\rho_{nw} \lambda_{nw} \boldsymbol{\kappa} \cdot \nabla\Phi_{nw})$ are wetting and non-wetting fluid phase mass fluxes [2], with $\lambda_\ell = k_{r\ell}/\mu_\ell$ the phase mobility, μ_ℓ the phase viscosity, $k_{r\ell}$ the phase relative permeability factor, $\boldsymbol{\kappa}$ the absolute permeability tensor, $\Phi_\ell = (p - \rho_\ell \boldsymbol{g} \cdot \boldsymbol{x})$ the phase potential, $\boldsymbol{x}$ the position vector in $\mathbb{R}^3$, $\ell = \{w, nw\}$;
- q_w and q_{nw} are mass source/sink per unit volume terms for the wetting and the non-wetting fluid phase, respectively;
- ∇, ∇^s, and $\nabla\cdot$ are the gradient, symmetric gradient, and divergence operator, respectively;
- the superposed dot, $(\dot{\bullet})$, indicates the derivative of quantity $(\bullet)$ with respect to time.

For the application of boundary conditions, let us introduce two disjoint partitions of the domain boundary such that $\Gamma = \overline{\Gamma_u^D \cup \Gamma_u^N} = \overline{\Gamma_f^D \cup \Gamma_f^N}$. Without loss of generality, consider homogeneous displacement boundary conditions $\boldsymbol{u} = \boldsymbol{0}$ on $\Gamma_u^D \times \mathbb{I}$ and homogeneous flux conditions $\boldsymbol{w}_w \cdot \boldsymbol{n} = \boldsymbol{w}_{nw} \cdot \boldsymbol{n} = 0$ on $\Gamma_f^N \times \mathbb{I}$, along with total traction conditions $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}^N$ on $\Gamma_u^N \times \mathbb{I}$ and pressure/saturation conditions $p = p^D$ and $s = s^D$ on $\Gamma_f^D \times \mathbb{I}$, where $\boldsymbol{n}$ denotes the outer normal vector for Γ. More complicated boundary conditions may be introduced as needed with modest changes to the discretization below. The formulation is completed by appropriate (i) initial conditions for $\boldsymbol{u}$, s, and p, and (ii) equations of state and constitutive equations to specify the following dependencies: $\phi = \phi(\boldsymbol{u}, p)$, $\rho_\ell = \rho_\ell(p)$, $\mu_\ell = \mu_\ell(p)$, and $k_{r\ell} = k_{r\ell}(s)$, with $\ell = \{w, nw\}$. For additional details on the adopted poromechanical model we refer the reader to [49].

3. Discretization. Several space discretization methods for the multiphase poromechanical IBVP have been proposed in the literature—see, e.g., [31, 23, 49] and references therein. In this work, the discrete form of (2.1) is obtained by combining a finite element (FE) method for the mechanical subproblem with a finite volume (FV) approach for the multiphase flow and transport subproblem. This choice is quite common when modeling nonlinear hydromechanical processes in subsurface formations characterized by highly heterogeneous hydrogeological properties, e.g., high-contrast permeability fields typically encountered in practical reservoir simulation [40, 28, 35, 20, 41].

Let $\boldsymbol{H}_0^1(\Omega)$ denote the Sobolev space of vector functions satisfying displacement homogeneous Dirichlet conditions over Γ_u^D and whose first derivatives belong to $L^2(\Omega)$, with $L^2(\Omega)$ the space of square integrable functions in Ω; let $\boldsymbol{\mathcal{U}}^h \subset \boldsymbol{H}_0^1(\Omega)$, $\mathcal{S}^h \subset L^2(\Omega)$, $\mathcal{P}^h \subset L^2(\Omega)$ denote finite-dimensional subspaces associated with a conforming triangulation $\mathcal{T}^h$ of the domain into nonoverlapping hexahedral cells; and let $\widehat{w}_\ell^\varepsilon$ denote a conservative numerical flux approximating the ℓ fluid phase mass flux across face ε in $\mathcal{E}^h$, namely the set of faces in $\mathcal{T}^h$, such that $\widehat{w}_\ell^\varepsilon \approx \int_\varepsilon \boldsymbol{w}_\ell \cdot \boldsymbol{n}_\varepsilon \, \mathrm{d}A$, with $\boldsymbol{n}_\varepsilon$ a unit normal vector defining the global face orientation.

Precisely, our space discretization employs (i) continuous piecewise trilinear finite elements for $\boldsymbol{\mathcal{U}}^h$, (ii) piecewise constant functions for $\mathcal{S}^h$ and $\mathcal{P}^h$, and (iii) a linear two-point flux approximation (TPFA) combined with a first-order upwinding strategy for the numerical flux $\widehat{w}_\ell^\varepsilon$ [2]. Using a fully implicit time marching scheme, with the subscript n indicating the discrete time level, the fully discrete mixed FE/FV variational statement of (2.1) is the following: find $\{\boldsymbol{u}_n^h, s_n^h, p_n^h\} \in \boldsymbol{\mathcal{U}}^h \times \mathcal{S}^h \times \mathcal{P}^h$ such that for all $\{\boldsymbol{\eta}^h, \psi^h, \chi^h\} \in \boldsymbol{\mathcal{U}}^h \times \mathcal{S}^h \times \mathcal{P}^h$,

$$(3.1a) \quad \mathcal{F}_u = \left(\nabla^s \boldsymbol{\eta}^h, \mathbf{C} : \nabla^s \boldsymbol{u}_n^h\right) - \left(\nabla \cdot \boldsymbol{\eta}^h, b p_n^h\right) - \left(\boldsymbol{\eta}^h, \rho_n \boldsymbol{g}\right) - \int_{\Gamma_u^N} \boldsymbol{\eta}^h \cdot \boldsymbol{t}_n^N \, \mathrm{d}A = 0,$$

$$(3.1b) \quad \mathcal{F}_s = \left(\psi^h, \frac{m_{w,n} - m_{w,n-1}}{\Delta t_n}\right) - \sum_{\varepsilon \in \mathcal{E}^h \setminus \mathcal{E}_f^{h,N}} [\![\psi^h]\!]_\varepsilon \widehat{w}_{w,n}^\varepsilon - (\psi^h, q_{w,n}) = 0,$$

$$(3.1c) \quad \mathcal{F}_p = \left(\chi^h, \frac{m_{nw,n} - m_{nw,n-1}}{\Delta t_n}\right) - \sum_{\varepsilon \in \mathcal{E}^h \setminus \mathcal{E}_f^{h,N}} [\![\chi^h]\!]_\varepsilon \widehat{w}_{nw,n}^\varepsilon - (\chi^h, q_{nw,n}) = 0,$$

where $n \in \{1, 2, \ldots\}$. The compact notation $(\bullet, \bullet)$ denotes the L^2-inner product of scalar, vector, or rank-2 tensor functions in $L^2(\Omega)$, $[L^2(\Omega)]^3$, or $[L^2(\Omega)]^{3\times3}$, as appropriate. In (3.1b), (3.1c), $\Delta t_n = (t_n - t_{n-1})$ is the time-step size; $\mathcal{E}_f^{h,N}$ is the set of faces belonging to the boundary Γ_f^N; and $[\![\bullet]\!]_\varepsilon$ indicates the jump of a quantity $(\bullet)$ across ε. For an internal face ε shared by cells K and L, $\boldsymbol{n}_\varepsilon$ pointing from K to L, $[\![\bullet]\!]_\varepsilon = ((\bullet)_{|L} - (\bullet)_{|K})$, with $(\bullet)_{|K}$ and $(\bullet)_{|L}$ the restriction of $(\bullet)$ on K and L, respectively. For a boundary lying on Γ_f^D, $\boldsymbol{n}_\varepsilon$ coincides with the outer normal to the domain boundary and the jump expression simply reads $[\![\bullet]\!]_\varepsilon = -(\bullet)_{|K}$.

Finally, introducing in (3.1) the expressions $\boldsymbol{u}_n^h = \sum_i u_{i,n} \boldsymbol{\eta}_i^h$, $s_n^h = \sum_j s_{j,n} \psi_j^h$, and $p_n^h = \sum_k p_{k,n} \chi_k^h$, with $\{\boldsymbol{\eta}_i^h\}$, $\{\psi_j^h\}$, and $\{\chi_k^h\}$ bases for $\boldsymbol{\mathcal{U}}^h$, $\mathcal{S}^h$, and $\mathcal{P}^h$, respectively, a standard Galerkin approach yields a system of nonlinear discrete equations

$$(3.2) \qquad F(\mathbf{x}_n) = \begin{pmatrix} F_u(\mathbf{x}_n) \\ F_s(\mathbf{x}_n) \\ F_p(\mathbf{x}_n) \end{pmatrix} = \boldsymbol{0}.$$

Here, vector $\mathbf{x}_n$ contains the nodal displacement $(u_{i,n})$, cell-centered saturation $s_{i,n}$, and cell-centered pressure $p_{i,n}$ coefficients that are used to expand $\boldsymbol{u}_n^h$, s_n^h, and p_n^h in terms of the respective basis functions at time level n.

Remark 3.1. Despite being routinely used in practical applications, the choice for the interpolation spaces $\boldsymbol{\mathcal{U}}^h$, $\mathcal{S}^h$, and $\mathcal{P}^h$ does not lead to an inf-sup stable discretization. In the incompressible limit, nonphysical spurious pressure modes can appear (i.e., for small permeability or when very small time steps are used). We address this issue by adding additional pressure jump stabilization terms to the mass balance equations following an approach proposed in [26]. For a detailed presentation on the adopted stabilization technique we refer the reader to [9].

4. Newton–Krylov solver. The nonlinear system (3.2) is solved by means of Newton's method, with a backtracking strategy added for robustness. The solution at time t_n is computed as follows. Given an initial guess $\mathbf{x}_n^0$, for $k = 0, 1, \ldots$, until convergence

$$(4.1) \qquad \begin{aligned} \text{solve} \quad & A(\mathbf{x}_n^{(k)})\Delta\mathbf{x} = -F(\mathbf{x}_n^{(k)}), \\ \text{set} \quad & \mathbf{x}_n^{(k+1)} = \mathbf{x}_n^{(k)} + \lambda\Delta\mathbf{x}, \end{aligned}$$

where $A(\mathbf{x}_n^{(k)}) = (\partial F/\partial\mathbf{x}_n)(\mathbf{x}_n^{(k)})$ is the Jacobian matrix associated with the nonlinear residual function F, and $\lambda \in (0,1]$ is an appropriately chosen line-search parameter. For convenience of notation, from now on we omit specifying that A is evaluated at $\mathbf{x}_n^{(k)}$. Clearly, at each nonlinear iteration k, the solution of a linear system with A is required.

The linearization of (3.2) produces a Jacobian system with an inherent 3×3 block structure

$$(4.2) \qquad A = \begin{pmatrix} A_{uu} & A_{us} & A_{up} \\ A_{su} & A_{ss} & A_{sp} \\ A_{pu} & A_{ps} & A_{pp} \end{pmatrix}.$$

This system has size proportional to the number of vertices (three displacement degrees of freedom per vertex) and cells (one saturation and one pressure degree of freedom per cell) in the computational mesh. For detailed expressions of the subblocks in A, we refer the reader to [49]. Briefly, we emphasize the properties of the three diagonal blocks that motivate choices in designing the preconditioning operator described in section 5. Specifically,

- A_{uu} is the elasticity block and has the structure of a discrete elliptic operator;
- A_{ss} is the saturation block that, in the absence of capillarity effects, has the structure of a discrete time-dependent hyperbolic problem;
- A_{pp} is the pressure block that, similar to the elastic block, has the structure of a discrete elliptic operator.

In this work, the linear system with matrix A is solved iteratively with generalized minimal residual (GMRES) [38], a Krylov subspace method designed for nonsymmetric systems. Since Krylov methods' practical convergence depends on the availability of an effective preconditioner, we introduce the preconditioning operator M and replace the linear system in (4.1) with the right preconditioned system,

$$(4.3) \qquad AM^{-1}\Delta\mathbf{y} = -F(\mathbf{x}_n^{(k)}),$$

where $\Delta\mathbf{x} = M^{-1}\Delta\mathbf{y}$. In the following section, we describe an algebraic method to construct M given a matrix A with the structure specified in (4.2).

5. Multigrid reduction. The idea of MGR has been around for a long time, tracing back to the work of Ries and Trottenberg [36, 37]. Recently, it has gained more attention through the work on multigrid reduction in time by Falgout et al. [16, 17]. MGR has also been applied successfully for problems in reservoir simulation and multiphase flow in porous media with phase transitions [8, 47]. A major advantage of the MGR approach is that it is an algebraic method and unlike geometric multigrid, it can be applied to general geometries and grid types. In this section, we first summarize the approach for the case of two-level reduction and then present the general multilevel reduction algorithm.

5.1. Two-grid reduction scheme. For a matrix A of size $N \times N$, we define a partition of the row indices of the matrix into C-points and F-points. The C-points play a role analogous to the points on a coarse grid, and the F-points belong to the set that is the complement of the C-points. It is important to note that this partitioning is different from the one normally used in standard multigrid methods, in which the F-points correspond to all points on the fine grid, i.e., the set of F-points contains the set of C-points. In multigrid reduction, the C-points and F-points belong to nonoverlapping sets. Following [16], using such CF-splitting we have

$$(5.1) \quad A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \begin{pmatrix} I_{FF} & 0 \\ A_{CF}A_{FF}^{-1} & I_{CC} \end{pmatrix} \begin{pmatrix} A_{FF} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I_{FF} & A_{FF}^{-1}A_{FC} \\ 0 & I_{CC} \end{pmatrix},$$

where I_{CC} and I_{FF} are identity matrices and $S = A_{CC} - A_{CF}A_{FF}^{-1}A_{FC}$ is the Schur complement. We can define the ideal interpolation and restriction operators by

$$(5.2) \quad P = \begin{pmatrix} -A_{FF}^{-1}A_{FC} \\ I_{CC} \end{pmatrix}, \quad R = \begin{pmatrix} -A_{CF}A_{FF}^{-1} & I_{CC} \end{pmatrix}.$$

Additionally, define the injection operator as $Q = \begin{pmatrix} I_{FF} \\ 0 \end{pmatrix}$. Then since $A_{FF} = Q^T A Q$ and $S = RAP$, it is simple to derive that

$$(5.3) \quad A^{-1} = P(RAP)^{-1}R + Q(Q^TAQ)^{-1}Q^T$$

and

$$(5.4) \quad 0 = I - A^{-1}A = I - P(RAP)^{-1}RA - Q(Q^TAQ)^{-1}Q^TA$$

$$(5.5) \quad = (I - P(RAP)^{-1}RA)(I - Q(Q^TAQ)^{-1}Q^TA)$$

$$(5.6) \quad = (I - Q(Q^TAQ)^{-1}Q^TA)(I - P(RAP)^{-1}RA),$$

where the equivalence occurs since $RAQ = Q^TAP = 0$. This identity defines the two-level multigrid method with the ideal Petrov–Galerkin coarse-grid correction $(RAP)^{-1}$ and the F-relaxation $Q(Q^TAQ)^{-1}Q^T$: (i) (5.4) is the additive MGR identity, and (ii) (5.5) and (5.6) are multiplicative identities with presmoothing and postsmoothing F-relaxation, respectively. However, constructing ideal interpolation and restriction operators is impractical. Similarly, computing the coarse-grid correction exactly is expensive, so we need to approximate these operators. In practice, MGR methods use a scalable solver such as AMG for the coarse-grid solve, and replace the ideal restriction and prolongation R and P with

$$(5.7) \quad \tilde{P} = \begin{pmatrix} W_p \\ I_{CC,} \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} W_r & I_{CC} \end{pmatrix},$$

where

$$(5.8) \qquad W_r \approx -A_{CF}A_{FF}^{-1}, \quad W_p \approx -A_{FF}^{-1}A_{FC}.$$

There are many ways to construct these approximations. One simple choice is to use an injection operator for restriction and a Jacobi approach for interpolation, namely

$$(5.9) \qquad W_r = 0, \quad W_p = -D_{FF}^{-1}A_{FC},$$

where $D_{FF} = \text{diag}(A_{FF})$. Then the coarse-grid operator $A_h = \tilde{R}A\tilde{P}$ can also be considered as an approximation to the Schur complement S. In addition to the choices in (5.9), one can also choose to use the Jacobi approach for restriction, that is, $W_r = -A_{CF}D_{FF}^{-1}$. Another option is to construct A_{FF}^{-1} using incomplete factorizations (ILU) or sparse approximate inverse techniques, such as sparse approximate inverse (SPAI) [22], factored sparse approximate inverse (FSAI) [19], or minimal residual (MR) [12]. Although these methods could provide a better approximation to A_{FF}^{-1}, and therefore better approximations for the restriction and interpolation operators, they tend to make these operators dense. The resulting coarse grid also becomes dense and unamenable to AMG. One can certainly apply a dropping strategy to keep such $\tilde{P}$ and $\tilde{R}$ sparse, but in practice, the potential improvement in performance using approximate inverse methods is usually offset by the cost to construct the approximation, which makes simple methods like Jacobi more appealing.

In general, we define the MGR operator with either presmoothing or postsmoothing F-relaxation by

$$(5.10) \qquad I - M_{MGR}^{-1}A = (I - \tilde{P}M_{CC}^{-1}\tilde{R}A)(I - M_{FF}^{-1}A),$$

$$(5.11) \qquad I - M_{MGR}^{-1}A = (I - M_{FF}^{-1}A)(I - \tilde{P}M_{CC}^{-1}\tilde{R}A),$$

where $M_{CC} = (\tilde{R}A\tilde{P})$ is the coarse-grid correction and M_{FF}^{-1} is the F-relaxation smoother. Additionally, similar to AMG methods, one can also apply a global smoothing step that extends to all the unknowns, not just the F-points. For the global smoother M_{glo}^{-1}, various methods, including (block) Jacobi, (block) Gauss–Seidel, or ILU, can be used. The inclusion of this step can help eliminate error modes that both the F-relaxation and coarse-grid correction may have missed. The application of the two-grid MGR scheme consisting of a global smoother and an F-relaxation followed by a coarse-grid correction can be summarized as shown in Algorithm 5.1.

Algorithm 5.1. Two-grid MGR preconditioner with presmoothing, $z = M_{MGR}^{-1}v$.

1: **function** MGR(A, v)
2: $\qquad z = M_{glo}^{-1}v$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Global relaxation
3: $\qquad z \leftarrow z + QM_{FF}^{-1}Q^T(v - Az)$ $\qquad\qquad\qquad\qquad$ ▷ F-relaxation
4: $\qquad r_C = \tilde{R}(v - Az)$ $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Restrict residual
5: $\qquad M_{CC}e_C = r_C$ $\qquad\qquad$ ▷ Solve coarse-grid error problem with AMG
6: $\qquad e = \tilde{P}e_C$ $\qquad\qquad\qquad$ ▷ Interpolate coarse error approximation
7: $\qquad z \leftarrow z + e$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ Apply correction
8: $\qquad$ **return** z
9: **end function**

Balancing the quality of the approximation to the Schur complement and the convergence of the coarse-grid solve is key to the success of MGR. One extreme is to

design a coarse grid that is perfectly suitable for AMG. Assuming, for example, that the block A_{CC} comes from a scalar elliptic PDE and A_{CC} is symmetric positive definite, then one can choose $W_p = W_r = 0$ and the coarse grid becomes $RAP = A_{CC}$. In this case, the convergence of the coarse-grid solve is optimal, but the approximation of the Schur complement is far from ideal, since the coarse grid takes into account neither any information from the F-points nor the coupling between the C- and F-points. At the other extreme, one can use the exact Schur complement as the coarse grid by choosing $W_r = -A_{CF}A_{FF}^{-1}$ and $W_p = 0$. However, because of the exact inversion of A_{FF}^{-1}, the coarse grid is dense. Furthermore, since the F-points and C-points actually represent equations obtained from the discretization of different continuous physical models, capturing the coupling between them on the coarse grid can lead to loss of ellipticity, which can make the coarse-grid solve with AMG ineffective. Thus, finding a good approximation of the Schur complement that is still amenable to AMG methods is essential.

Remark 5.1. The appeal of the MGR approach is that it provides a general framework for choosing the coarse/fine grids, the interpolation and restriction operators, and the solvers for the F-relaxation and coarse-grid correction. As an example, it was shown in [8, 47] that one can recast any CPR-AMG strategy [10, 21, 29, 32, 39, 43, 51] or block preconditioner [7] used in reservoir simulation as a particular variant of the two-grid MGR reduction scheme by appropriately defining the different components of the algorithm, namely prolongation, restriction, and smoothing operators.

5.2. A general multilevel MGR algorithm. One can replace the coarse-grid solve in Algorithm 5.1 with a two-level MGR scheme and apply the method recursively to obtain a multilevel MGR algorithm. The general application of the MGR V-cycle with global smoothing is summarized in Algorithm 5.2, where the hierarchy of coarse-grid operators, i.e., $A_{l+1} = \tilde{R}_l A_l \tilde{P}_l$, is assumed to be computed for each level l.

Algorithm 5.2. General multilevel MGR preconditioner, $z = M_{l,MGR}^{-1}v$.

1: **function** V_MGR(A_l, v_l)
2:     **if** l is the coarsest level **then**
3:         $A_l z_l = v_l$ $\triangleright$ Solve coarse-grid error problem with AMG
4:     **else**
5:         $z_l = M_{l,glo}^{-1}v_l$ $\triangleright$ Global relaxation
6:         $z_l \leftarrow z_l + Q_l M_{l,FF}^{-1} Q_l^T(v_l - Az_l)$ $\triangleright$ F-relaxation
7:         $r_{l+1} = \tilde{R}_l(v_l - A_l z_l)$ $\triangleright$ Restrict residual
8:         $e_{l+1} = \text{MGR}(A_{l+1}, r_{l+1})$ $\triangleright$ Recursion
9:         $e_l = \tilde{P}_l e_{l+1}$ $\triangleright$ Interpolate coarse error approximation
10:       $z_l \leftarrow z_l + e_l$ $\triangleright$ Apply correction
11:     **end if**
12:     **return** z_l
13: **end function**

Based on Algorithm 5.2, W- and F-cycle versions of the MGR algorithm can also be defined [37]. Note that the Schur complement S is approximated by the triple product $RAP = A_{CC} - A_{CF}D_{FF}^{-1}A_{FC}$ in the classical two-grid reduction scheme in Algorithm 5.1. Even though we have introduced a sparse approximation to S by replacing A_{FF} with its diagonal D_{FF}, i.e., $A_{FF}^{-1} \approx D_{FF}^{-1}$, in a multilevel reduction scheme, the coarse grid can still become dense or unsuitable for standard AMG be-

cause the correction term $A_{cor} = A_{CF} D_{FF}^{-1} A_{FC}$ involves a matrix-matrix product.

In this work, we develop a dropping strategy for A_{cor} to keep the coarse grid sparse as well as suitable for AMG. One approach is to drop all entries of A_{cor} that are smaller than a prescribed tolerance. Here, we use a different strategy based on a maximum number of nonzero values per row. Specifically, we choose to keep only $N_{\max}$ entries with largest absolute values on each row. To preserve at least some information of the first level of reduction, however, we always keep the diagonals of the subblocks in A_{cor}. For instance, in a three-level reduction scheme, in the first-level reduction, A_{cor} has 2×2 block structure, and applying maximum dropping (i.e., using an extremely large tolerance or $N_{\max} = 0$), A_{cor} is still a 2×2 block matrix, whose subblocks are diagonal matrices. Using this dropping strategy results in a non-Galerkin coarse grid

$$(5.12) \qquad S = A_{CC} - \mathbf{G}(A_{CF} D_{FF}^{-1} A_{FC}),$$

where $\mathbf{G}$ is a sparsifying operator that performs one of the aforementioned dropping strategies. So far we only assume that a CF-splitting of the rows is given. How to choose such a splitting is dependent on the problem and it is up to the user to make the decision. However, as a general principle, it is usually a good idea to choose a CF-splitting so that the final coarse grid corresponds to the variable associated with an elliptic equation, e.g., pressure, since we want to solve the coarse grid using an efficient method such as standard AMG. In the next section, we show how to choose an appropriate CF-splitting at each level of reduction for our multiphase poromechanical problem.

5.3. MGR for multiphase poromechanics. We propose a three-level MGR reduction scheme to precondition the Jacobian matrix (4.2). For the first level of reduction, we aim at decoupling the mechanics subproblem from the flow. Therefore, we assign all the displacement unknowns as F-points, while both saturation and pressure unknowns are labeled as C-points. This leads to the following partitioning:

$$(5.13) \qquad A = \left(\begin{array}{c|cc} A_{uu} & A_{us} & A_{up} \\ \hline A_{su} & A_{ss} & A_{sp} \\ A_{pu} & A_{ps} & A_{pp} \end{array} \right) \quad \begin{array}{c} F \\ C \\ C \end{array} .$$

Then $A_{FF} \equiv A_{uu}$ and the coarse grid A_{CC}, which corresponds to the flow subproblem, has the 2×2 block structure

$$(5.14) \qquad A_{CC} = \begin{pmatrix} A_{ss} & A_{sp} \\ A_{ps} & A_{pp} \end{pmatrix}.$$

For the F-relaxation step, we need to solve the elasticity problem involving the elliptic operator A_{uu}. Here, we use one AMG V-cycle. Because of the vectorial nature of the elasticity operator, this is the most expensive part of the setup phase. Also, given the global system size, we ignore the first-level global relaxation step. Using the interpolation and restriction operators specified in (5.9) combined with the dropping strategy defined in (5.12) yields the following first-level coarse grid:

$$(5.15) \qquad \begin{aligned} S_1 &= \begin{pmatrix} A_{ss} & A_{sp} \\ A_{ps} & A_{pp} \end{pmatrix} - \mathbf{G}\left(\begin{pmatrix} A_{su} \\ A_{pu} \end{pmatrix} D_{uu}^{-1} \begin{pmatrix} A_{us} & A_{up} \end{pmatrix} \right) \\ &= \begin{pmatrix} \tilde{A}_{ss} & \tilde{A}_{sp} \\ \tilde{A}_{ps} & \tilde{A}_{pp} \end{pmatrix}. \end{aligned}$$

For our multiphase poromechanics problem, we use $N_{\max} = 4$ for $\mathbf{G}$. Again, we emphasize the flexibility of our framework as it allows for experimenting with different choices of $\mathbf{G}$. For example, choosing an appropriate $\mathbf{G}$, we can mimic the fixed-stress preconditioner developed in [49].

For the second reduction step, we label saturation unknowns as F-points and pressure unknowns as C-points in S_1:

$$(5.16) \qquad S_1 = \left(\begin{array}{c|c} \tilde{A}_{ss} & \tilde{A}_{sp} \\ \hline \tilde{A}_{ps} & \tilde{A}_{pp} \end{array} \right) \quad \begin{array}{c} F \\ C \end{array} .$$

Again, using interpolation and restriction operators as in (5.9), we obtain the second-level Schur complement

$$(5.17) \qquad S_2 = \tilde{A}_{pp} - \tilde{A}_{ps} D_{ss}^{-1} \tilde{A}_{sp},$$

where $D_{ss} = \text{diag}(\tilde{A}_{ss})$. Here, no additional sparse approximation is needed after this reduction since the flow part is relatively small compared to the elasticity block and the coarse grid generated in (5.17) is still well-suited for AMG. We note that the reduction in (5.17) is essentially a CPR approach that is embedded within a multigrid reduction framework. Thus, the MGR approach is closely related to block factorization approaches with approximate Schur complements. In MGR, the Schur complement is computed by the Galerkin triple product RAP as shown in section 5.1, and the approximation to the Schur complement is obtained through approximating the interpolation and restriction operators. For instance, the MGR approach is equivalent to the algebraic block factorization preconditioner in [7] for the choices of $W_r = -A_{FC} D_{FF}^{-1}$ and $W_p = 0$, instead of those specified in (5.9). In other block factorization and CPR variants, approximations to the Schur complement are usually derived directly using knowledge of the physics. For example, in a *Quasi-IMPES reduction* scheme, the block $\tilde{A}_{ps}$ is also replaced by its diagonal $D_{ps} = \text{diag}(\tilde{A}_{ps})$, leading to an approximate Schur complement of the form

$$(5.18) \qquad S_2^{(CPR)} = \tilde{A}_{pp} - D_{ps} D_{ss}^{-1} \tilde{A}_{sp}.$$

This approximation ensures that the matrix sparsity pattern of A_{pp} coming from the original finite volume stencil is preserved and $S_2^{(CPR)}$ is near elliptic. In the MGR framework, $S_2^{(CPR)}$ cannot be derived using a Galerkin approach for practical choices of $\tilde{R}$ and $\tilde{P}$, and one needs to use a non-Galerkin dropping strategy similar to the one introduced in (5.12) to achieve a specific sparsity pattern.

In the solve phase, at the second level, the F-relaxation involving the $\tilde{A}_{ss}$ block is done using a simple Jacobi relaxation. However, the second-level global smoothing step is required to reduce the error associated with the hyperbolic component of the flow subproblem. Indeed, the global smoothing plays a key role particularly in the later stage of the simulation when the pressure field approaches steady-state conditions and the multiphase flow and transport process transitions to an advection dominated regime. The need for a robust global smoother will become clear through numerical results presented in the next section.

Remark 5.2. Even though we formally present the multigrid reduction framework for the 3×3 system in field-ordered form, in our implementation, the input matrix has interleaved ordering for saturation and pressure. This choice produces a sparsity pattern in which dense 2×2 blocks appear for the first-level coarse grid in (5.15).

Block versions of relaxation or incomplete factorization preconditioners are therefore appealing, as dense multiplication and inversion operations can be applied to the small blocks.

Remark 5.3. Common strategies for the second-level global smoothing step include block relaxation methods (e.g., Jacobi, Gauss–Seidel) or incomplete factorizations (e.g., ILU(k), ILUT). In this work, we explore two options. The first option uses several sweeps of hybrid block Gauss–Seidel (HBGS) [3]. The second option uses one sweep of processor-local, pointwise ILU(k) [11].

Remark 5.4. Since the flow is driven by pressure in the physical model, one might be inclined to use a CPR approach that reflects an implicit pressure explicit saturation (IMPES) splitting scheme. In this scenario, one first solves for pressure, and then saturation is updated using the new pressure field. This ordering can easily be accomplished using the current MGR framework. In Algorithm 5.2, even though we present a presmoothing scheme in which global relaxation is followed by F-relaxation and coarse-grid correction, our implementation also supports postsmoothing, and one has the option to do F-relaxation and global relaxation after coarse-grid correction. For our numerical examples below, we do not notice a significant difference in convergence between the pre- and postsmoothing approaches.

6. Numerical results. We perform numerical experiments to test the performance of the MGR preconditioner on two problems: (1) a weak scaling study for a simple synthetic configuration, and (2) a strong scaling study using a realistic, highly heterogeneous reservoir based on the SPE10 [13] example. Both examples have been designed as community benchmark problems and exhibit tight coupling between displacement, pressure, and saturation fields. Problem specifications are described in detail in [49] and so are only briefly reported below.

In this study, the simulator is provided by *Geocentric*, which utilizes the *deal*.II finite element library [4] for discretization functionality. It also provides a direct interface with MGR, which is implemented as a separate solver in *hypre* [18]. All the numerical experiments were run on *Quartz*, a cluster at the Lawrence Livermore Computing Center with 1344 nodes containing two Intel Xeon E5-2695 18-core processors sharing 128 GiB of memory on each node with Intel Omni-Path interconnects between nodes. We use pure MPI-based parallelism.

For the elastic block A_{uu}, we use one V-cycle of BoomerAMG [25], with an unknown approach for a system of three PDEs, with one level of aggressive coarsening, one sweep of hybrid forward l_1-Gauss–Seidel [3] for the down cycle, and one sweep of hybrid backward l_1-Gauss–Seidel for the up cycle. The coarsest grid is solved directly with Gaussian elimination. The MGR coarse solve in (5.17) also uses BoomerAMG with the same smoother configuration, but for a scalar problem and a hybrid modified independent set (HMIS) coarsening strategy [15]. For the global smoother, we use one step of processor-local, pointwise ILU(1).

6.1. Staircase benchmark. The configuration of the first test problem is illustrated in Figure 1. A highly permeable channel winds its way in a "staircase" fashion through a lower-permeability host rock. A denser, wetting phase is injected through a well at the top corner, leading to a saturation plume driven by gravity and pressure that migrates towards a production well in the lower corner. The whole system is deformable and exhibits significant poromechanical coupling. Visualizations of the resulting pressure and deformation fields have been omitted for the sake of brevity. A detailed specification of mesh geometry, material properties, and boundary condi-
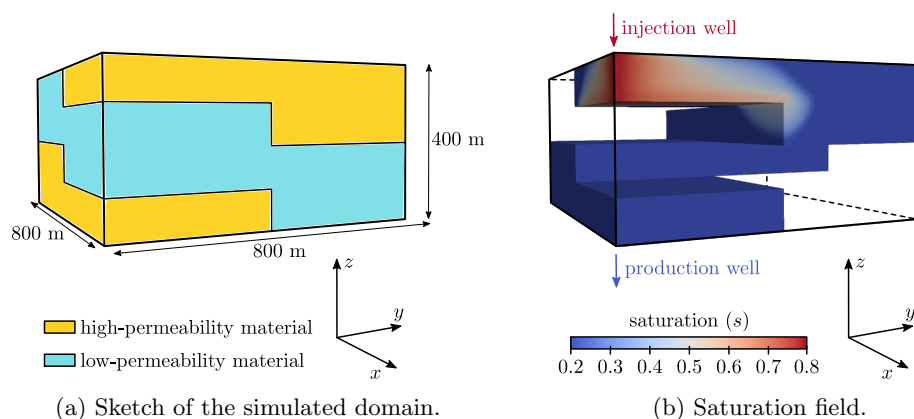
(a) Sketch of the simulated domain.   (b) Saturation field.

FIG. 1. *Staircase benchmark, showing basic geometry and resulting saturation field within the high-permeability channel at $t = 92$ days.*

TABLE 1
Weak scaling performance for the staircase example.

Cores	DoF	Newton iterations per time step (avg.)	GMRES iterations per Newton (avg.)	Setup phase [s]	Solve phase [s]	Iteration increase	Time increase
2	680,419	3.1	10.2	5.25	6.26	1.00	1.00
16	5,342,147	3.8	11.3	8.36	9.92	1.10	1.59
128	42,338,179	4.7	13.7	9.20	19.17	1.34	2.46
1,024	337,121,027	6.1	15.9	11.54	25.31	1.56	3.20

tions can be found in [49]. Note that compared to the setup used in [49], a larger domain is considered here to enable simulation on higher resolution computational grids. The highest grid resolution in [49] is limited by the physical well radius used in the Peaceman well model [34].

Table 1 shows the results for a weak scaling study using the staircase example. We keep the number of degrees of freedom per core approximately constant at 340k and increase the number of cores from 2 to 1,024. The global problem size grows 8^3 times from 680k to 337M. We report the average number of Newton iterations per time step, the average number of GMRES iterations per Newton iteration, the average setup and solve time per Newton iteration, the factor of increase in GMRES iteration count, and the factor of increase in the average time per Newton iteration. Due to the inherent nonlinearity, we observe an increase in the number of Newton iterations per time step as the mesh is refined. The average number of GMRES iterations per Newton step, however, only experiences a modest growth as desired. The growth in the number of iterations is due to the fact that we are using a single V-cycle for the elasticity block A_{uu} in the first F-relaxation step. We can use a more complex smoother in place of the hybrid l_1-Gauss–Seidel solves or increase the number of V-cycles to drive down the number of iterations, but that will come at the expense of run-time performance. In general, we find that the l_1-Gauss–Seidel smoother strikes a good balance between iteration counts and run time. Similar to the number of iterations, the total run time, including both the set-up and solve phases, also exhibits some growth, but again, the result is quite satisfactory even for large core counts. The increase in the run time can be attributed to communication costs in the MGR set-up and solve phases. The most
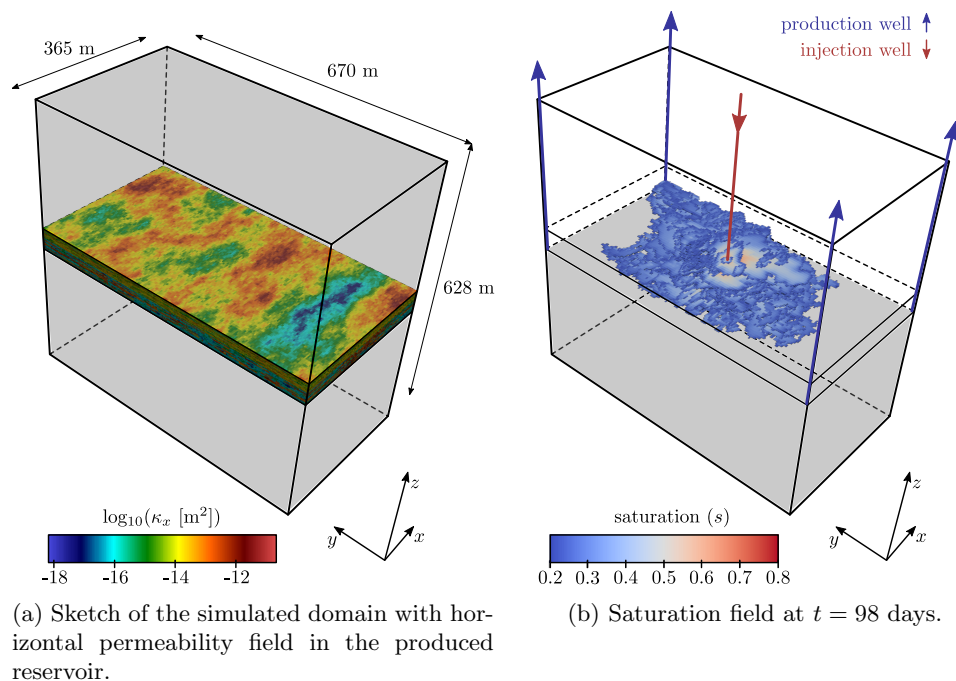
(a) Sketch of the simulated domain with horizontal permeability field in the produced reservoir.

(b) Saturation field at $t = 98$ days.

FIG. 2. *SPE*10-*based benchmark: The original SPE*-10 *reservoir is embedded in a larger poromechanical domain to provide realistic mechanical boundary conditions.*

notable increase is in the solve phase from 16 to 128 cores, because these 128 cores are distributed over 4 computing nodes, while all 16 cores reside on the same node, and off-node communication is expensive. Overall, however, the MGR framework provides a good platform for scalable performance.

6.2. SPE10-based benchmark. We also perform a strong scaling study on a more realistic benchmark problem (Figure 2) derived from the second model of the SPE10 Comparative Solution Project [13]. The original SPE10 permeability and porosity fields are now treated as a poromechanical medium. These geostatistically generated fields exhibit both severe heterogeneity and anisotropy. In the current poromechanical benchmark, the reservoir itself is also embedded in a larger domain— with caprock and underburden—to provide more realistic boundary conditions. Water is injected through a central well, while fluids are produced from four wells at the corners of the domain. Mesh, material property, and boundary condition specifications are reported in [49]. Note that the well control conditions differ from the original SPE10 model to avoid well impacts on the linear solver. The treatment of well degrees of freedom within the linear solver is a critical issue, but is deliberately not addressed in the current contribution. We remark, however, that the MGR approach provides a flexible framework for treating this additional complexity.

The resulting discrete problem has 16.7M degrees of freedom. We keep the problem size fixed and divide the work across an increasing number of compute cores. The results are shown in Table 2. Again, we observe only minor growth in the number of GMRES iterations with larger core counts. Similar to the weak scaling case, the reason for this growth is the use of a hybrid l_1-Gauss–Seidel smoother in AMG solves for the elasticity block and the coarse grid. Good overall timing efficiency is also

TABLE 2
Strong scaling performance for the SPE10-based problem.

Cores	DoF/core	Newton iterations per time step (avg.)	GMRES iterations per Newton (avg.)	Setup phase [s]	Solve phase [s]	Overall time [s]	Efficiency
36	464,738	6.5	21.0	15.60	44.2	20,290	1.00
72	232,369	6.5	21.5	7.29	22.7	10,160	1.00
144	116,184	6.5	22.5	4.40	11.7	5,470	0.93
288	58,092	6.5	23.2	2.48	6.2	2,941	0.86
576	29,046	6.5	24.3	2.58	3.2	1,955	0.65

achieved up to 288 cores. For 576 cores, even though we still get good efficiency for the solve phase, there is a noticeable increase in the setup time because the problem size on each core becomes very small, i.e., about 17k total and less than 6k degrees of freedom for the elastic block and the coarse grid, respectively. Consequently, the majority of the time is spent in communication while not much computation is performed. However, the results still indicate that one can use the proposed framework with a large number of processors to efficiently reduce the long simulation time for challenging problems with highly heterogeneous media.

6.3. Effect of global smoother. As we have mentioned earlier, the performance of MGR is dependent on the effectiveness of the solvers for each component of the algorithm. In general, changing the configuration for one component, e.g., smoother choices for the F-relaxation or coarsening strategies for the coarse-grid AMG solve, would result in a different number of GMRES iterations. However, the effect could also be quite subtle and not manifest itself until the underlying property of the problem changes. For multiphase poromechanics simulations, early times are typically dominated by elliptic effects associated with the pressure and displacement fields, while at late times the hyperbolic effects associated with the saturation field become significant.

Here we explore the effectiveness of different global smoothers on the multiphase flow system as the simulation progresses. The first option uses three sweeps of HBGS, and the second option uses a single sweep of processor-local ILU(1). As one can see from Figure 3a, there is no apparent difference between the two smoothers until about 45 days of injection, when the number of iterations for the HBGS method increases sharply and continues to stay high. In contrast, ILU(1) is less sensitive. Even though the number of iterations also rises slightly around 85 days, it starts to decrease for the last period at the end of the simulation. We also plot the total time of HBGS(3) against ILU(1) in Figure 3b. It is clear that even though ILU(1) takes slightly more time in the beginning (mainly due to higher setup cost), the tradeoff is worthwhile thanks to its robustness, which leads to a modest reduction in total time for the whole simulation. This observation is confirmed by the widespread use of incomplete factorization smoothers in the reservoir simulation community.

7. Conclusion. In this work, we have presented an algebraic framework based on multigrid reduction for solving the linear system that comes from discretizing and linearizing the conservation equations governing multiphase flow coupled with poromechanics. This framework is flexible and allows us to construct different preconditioners based on different choices for CF-splitting strategies, interpolation and restriction operators, as well as solvers and smoothers. We have also developed a dropping strategy for computing the reduction onto the coarse grid within the MGR
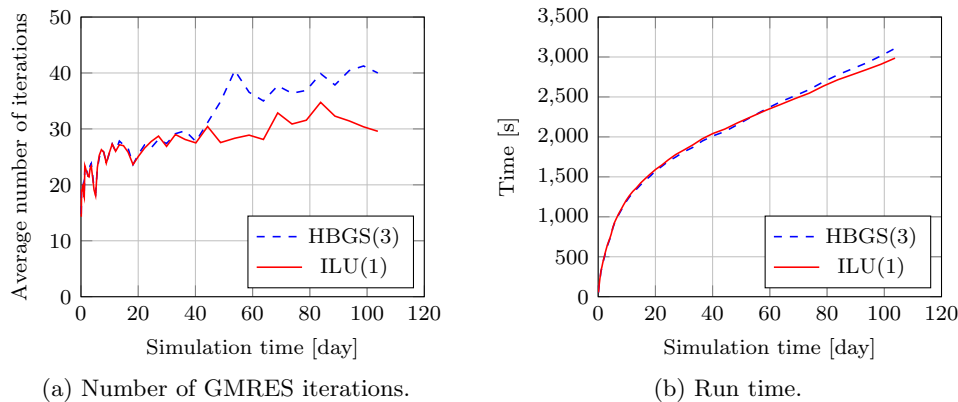
(a) Number of GMRES iterations.   (b) Run time.

FIG. 3. *Effects of the global smoother at different time steps for the hybrid block Gauss-Seidel and processor-local ILU(1).*

V-cycle that captures the coupling between mechanics and flow and reduces the operator complexity at the same time. This results in an algebraic preconditioner that is robust and scalable for realistic and large-scale simulation of multiphase poromechanics.

Regarding future work, a number of improvements to the MGR framework could be explored. For example, constructing good approximations to the ideal interpolation and restriction operators that have low complexity remains a significant challenge. Also, it is unclear how one can choose an optimal coarse grid that is representative of the fine-grid system and at the same time still amenable to AMG in a multilevel reduction setting. Thus, better strategies for computing the non-Galerkin coarse grid introduced in this work are needed to improve robustness of the framework. Finally, since MGR is designed to accommodate a wide range of coupled systems, we are looking into extending the approach to solve problems with nonisothermal flow and fractured media.

REFERENCES

[1] J. H. ADLER, F. J. GASPAR, X. HU, C. RODRIGO, AND L. T. ZIKATANOV, *Robust block preconditioners for Biot's model*, in Domain Decomposition Methods in Science and Engineering XXIV. DD 2017, P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, and O. B. Widlund, eds., Lect. Notes Comput. Sci. Eng. 125, Springer, Cham, 2018, pp. 3–16, https://doi.org/10.1007/978-3-319-93873-8_1.

[2] K. AZIZ AND A. SETTARI, *Petroleum Reservoir Simulation*, Elsevier Applied Science Publishers, London, UK, 1979.

[3] A. H. BAKER, R. D. FALGOUT, T. V. KOLEV, AND U. M. YANG, *Multigrid smoothers for ultraparallel computing*, SIAM J. Sci. Comput., 33 (2011), pp. 2864–2887, https://doi.org/10.1137/100798806.

[4] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II—A general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), 24, https://doi.org/10.1145/1268776.1268779.

[5] L. BERGAMASCHI, M. FERRONATO, AND G. GAMBOLATI, *Novel preconditioners for the iterative solution to FE-discretized coupled consolidation equations*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 2647–2656, https://doi.org/10.1016/j.cma.2007.01.013.

[6] L. BERGAMASCHI AND Á. MARTÍNEZ, *RMCP: Relaxed mixed constraint preconditioners for saddle point linear systems arising in geomechanics*, Comput. Methods Appl. Mech. Engrg., 221–222 (2012), pp. 54–62, https://doi.org/10.1016/j.cma.2012.02.004.

[7] Q. M. Bui, H. C. Elman, and J. D. Moulton, *Algebraic multigrid preconditioners for multiphase flow in porous media*, SIAM J. Sci. Comput., 39 (2017), pp. S662–S680, https://doi.org/10.1137/16M1082652.

[8] Q. M. Bui, L. Wang, and D. Osei-Kuffuor, *Algebraic multigrid preconditioners for two-phase flow in porous media with phase transitions*, Adv. Water Resour., 114 (2018), pp. 19–28, https://doi.org/10.1016/j.advwatres.2018.01.027.

[9] J. T. Camargo, J. A. White, and R. I. Borja, *A Macroelement Stabilization for Multiphase Poromechanics*, preprint, 2019, https://arxiv.org/abs/1909.08590v1.

[10] H. Cao, H. A. Tchelepi, J. R. Wallis, and H. E. Yardumian, *Parallel scalable unstructured CPR-type linear solver for reservoir simulation*, in Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, TX, 2005, https://doi.org/10.2118/96809-MS.

[11] E. Chow and A. Patel, *Fine-grained parallel incomplete LU factorization*, SIAM J. Sci. Comput., 37 (2015), pp. C169–C193, https://doi.org/10.1137/140968896.

[12] E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023, https://doi.org/10.1137/s1064827594270415.

[13] M. A. Christie and M. J. Blunt, *Tenth SPE Comparative Solution Project: A comparison of upscaling techniques*, SPE Reserv. Eval. Eng., 4 (2001), pp. 308–317, https://doi.org/10.2118/72469-PA.

[14] O. Coussy, *Poromechanics*, Wiley, Chichester, UK, 2004.

[15] H. De Sterck, U. M. Yang, and J. J. Heys, *Reducing complexity in parallel algebraic multigrid preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1019–1039, https://doi.org/10.1137/040615729.

[16] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661, https://doi.org/10.1137/130944230.

[17] R. D. Falgout, T. A. Manteuffel, B. O'Neill, and J. B. Schroder, *Multigrid reduction in time for nonlinear parabolic problems: A case study*, SIAM J. Sci. Comput., 39 (2017), pp. S298–S322, https://doi.org/10.1137/16M1082330.

[18] R. D. Falgout and U. M. Yang, *hypre: A library of high performance preconditioners*, in Computational Science — ICCS 2002, P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, eds., Lecture Notes in Comput. Sci. 2331, Springer, Berlin, Heidelberg, 2002, pp. 632–641, https://doi.org/10.1007/3-540-47789-6_66.

[19] M. Ferronato, C. Janna, and G. Pini, *A generalized Block FSAI preconditioner for nonsymmetric linear systems*, J. Comput. Appl. Math., 256 (2014), pp. 230–241, https://doi.org/10.1016/j.cam.2013.07.049.

[20] T. T. Garipov, M. Karimi-Fard, and H. A. Tchelepi, *Discrete fracture model for coupled flow and geomechanics*, Comput. Geosci., 20 (2016), pp. 149–160, https://doi.org/10.1007/s10596-015-9554-z.

[21] S. Gries, K. Stüben, G. L. Brown, D. Chen, and D. A. Collins, *Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations*, SPE J., 19 (2014), pp. 726–736, https://doi.org/10.2118/163608-PA.

[22] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853, https://doi.org/10.1137/s1064827594276552.

[23] J. B. Haga, H. Osmes, and H. P. Langtangen, *On the causes of pressure oscillations in low-permeable and low-compressible porous media*, Int. J. Numer. Anal. Methods Geomech., 36 (2012), pp. 1507–1522, https://doi.org/10.1002/nag.1062.

[24] J. B. Haga, H. Osmes, and H. P. Langtangen, *A parallel block preconditioner for large-scale poroelasticity with highly heterogeneous material parameters*, Comput. Geosci., 16 (2012), pp. 723–734, https://doi.org/10.1007/s10596-012-9284-4.

[25] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, https://doi.org/10.1016/S0168-9274(01)00115-5.

[26] N. Kechkar and D. J. Silvester, *Analysis of locally stabilized mixed finite element methods for the Stokes problem*, Math. Comp., 58 (1992), pp. 1–10, https://doi.org/10.1090/S0025-5718-1992-1106973-X.

[27] J. Kim, H. A. Tchelepi, and R. Juanes, *Stability, accuracy, and efficiency of sequential methods for coupled flow and geomechanics*, SPE J., 16 (2011), pp. 249–262, https://doi.org/10.2118/119084-PA.

[28] J. Kim, H. A. Tchelepi, and R. Juanes, *Rigorous coupling of geomechanics and multiphase flow with strong capillarity*, SPE J., 18 (2013), pp. 1123–1139, https://doi.org/10.2118/141268-PA.

[29] 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, https://doi.org/10.1137/s106482750240443x.

[30] J. J. Lee, K.-A. Mardal, and R. Winther, *Parameter-robust discretization and preconditioning of Biot's consolidation model*, SIAM J. Sci. Comput., 39 (2017), pp. A1–A24, https://doi.org/10.1137/15m1029473.

[31] R. W. Lewis and B. A. Schrefler, *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*, 2nd ed., Wiley, Chichester, UK, 1998.

[32] H. Liu, K. Wang, and Z. Chen, *A family of constrained pressure residual preconditioners for parallel reservoir simulations*, Numer. Linear Algebra Appl., 23 (2016), pp. 120–146, https://doi.org/10.1002/nla.2017.

[33] A. Mikelić and M. F. Wheeler, *Convergence of iterative coupling for coupled flow and geomechanics*, Comput. Geosci., 17 (2013), pp. 455–461, https://doi.org/10.1007/s10596-012-9318-y.

[34] D. W. Peaceman, *Interpretation of well-block pressures in numerical reservoir simulation*, SPE J., 18 (1978), pp. 183–194, https://doi.org/10.2118/6893-PA.

[35] J.-H. Prévost, *Two-way coupling in reservoir-geomechanical models: Vertex-centered Galerkin geomechanical model cell-centered and vertex-centered finite volume reservoir models*, Internat. J. Numer. Meth. Engrg., 98 (2014), pp. 612–624, https://doi.org/10.1002/nme.4657.

[36] M. Ries and U. Trottenberg, *MGR-Ein blitzschneller elliptischer Löser*, Preprint 277 SBF 72, Universität Bonn, 1979.

[37] M. Ries, U. Trottenberg, and G. Winter, *A note on MGR methods*, Linear Algebra Appl., 49 (1983), pp. 1–26, https://doi.org/10.1016/0024-3795(83)90091-5.

[38] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869, https://doi.org/10.1137/0907058.

[39] R. Scheichl, R. Masson, and J. Wendebourg, *Decoupling and block preconditioning for sedimentary basin simulations*, Comput. Geosci., 7 (2003), pp. 295–318, https://doi.org/10.1023/b:comg.0000005244.61636.4e.

[40] A. Settari and F. M. Mourits, *A coupled reservoir and geomechanical simulation system*, SPE J., 3 (1998), pp. 219–226, https://doi.org/10.2118/50939-PA.

[41] R. R. Settgast, P. Fu, S. D. C. Walsh, J. A. White, C. Annavarapu, and F. J. Ryerson, *A fully coupled method for massively parallel simulation of hydraulically driven fractures in 3-dimensions*, Int. J. Numer. Anal. Methods Geomech., 41 (2017), pp. 627–653, https://doi.org/10.1002/nag.2557.

[42] K. Stüben, *A review of algebraic multigrid*, J. Comput. Appl. Math., 128 (2001), pp. 281–309, https://doi.org/10.1016/S0377-0427(00)00516-1.

[43] K. Stüben, T. Clees, H. Klie, B. Lu, and M. F. Wheeler, *Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation*, in Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, 2007, https://doi.org/10.2118/105832-MS.

[44] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic Press, Orlando, FL, 2001.

[45] J. R. Wallis, *Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration*, in Proceedings of the SPE Reservoir Simulation Symposium, San Francisco, CA, 1983, https://doi.org/10.2118/12265-MS.

[46] J. R. Wallis, R. P. Kendall, and T. E. Little, *Constrained residual acceleration of conjugate residual methods*, in Proceedings of the SPE Reservoir Simulation Symposium, Dallas, TX, 1985, https://doi.org/10.2118/13536-MS.

[47] L. Wang, D. Osei-Kuffuor, R. D. Falgout, I. D. Mishev, and J. Li, *Multigrid reduction for coupled flow problems with application to reservoir simulation*, in Proceedings of the SPE Reservoir Simulation Conference, Montgomery, TX, 2017, https://doi.org/10.2118/182723-MS.

[48] J. A. White and R. I. Borja, *Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics*, Comput. Geosci., 15 (2011), pp. 647–659, https://doi.org/10.1007/s10596-011-9233-7.

[49] J. A. White, N. Castelletto, S. Klevtsov, Q. M. Bui, D. Osei-Kuffuor, and H. A. Tchelepi, *A two-stage preconditioner for multiphase poromechanics in reservoir simulation*, Comput. Methods Appl. Mech. Engrg., 357 (2019), 112575, https://doi.org/10.1016/j.cma.2019.112575.

[50] J. A. WHITE, N. CASTELLETTO, AND H. A. TCHELEPI, *Block-partitioned solvers for coupled poromechanics: A unified framework*, Comput. Methods Appl. Mech. Engrg., 303 (2016), pp. 55–74, https://doi.org/10.1016/j.cma.2016.01.008.

[51] Y. ZHOU, Y. JIANG, AND H. A. TCHELEPI, *A scalable multistage linear solver for reservoir models with multisegment wells*, Comput. Geosci., 17 (2013), pp. 197–216, https://doi.org/10.1007/s10596-012-9324-0.