IMPLICIT–EXPLICIT SCHEME WITH MULTISCALE VANKA TWO-GRID SOLVER FOR HETEROGENEOUS UNSATURATED POROELASTICITY[*]

MARIA VASILYEVA[†], BEN S. SOUTHWORTH[‡], YUNHUI HE[§], AND MIN WANG[§]

Abstract. We consider a coupled nonlinear system of equations that describe unsaturated flow in heterogeneous poroelastic media. For the numerical solution, we use a finite-element approximation in space and present an efficient multiscale two-grid solver for solving the coupled system of equations. The proposed two-grid solver contains two main parts: (i) an accurate coarse-grid approximation based on local spectral spaces, and (ii) coupled smoothing iterations based on an overlapping multiscale Vanka method. A Vanka smoother and local spectral coarse grids come with significant computational cost in the setup phase. To avoid constructing a new solver for each time step and/or nonlinear iteration, we utilize an implicit–explicit integration scheme in time, where we partition the nonlinear operator as a sum of linear and nonlinear parts. In particular, we construct an implicit linear approximation of the stiff components that remains fixed across all time, while treating the remaining nonlinear residual explicitly. This allows us to construct a robust two-grid solver offline and utilize it for fast and efficient online time integration. A linear stability analysis of the proposed novel coupled scheme is presented based on the representation of the system as a two-step scheme. We show that the careful decomposition of linear and nonlinear parts guarantees a linearly stable scheme. A numerical study is presented for a nonlinear coupled test problem of unsaturated flow in heterogeneous poroelastic media. We demonstrate the robustness of the two-grid solver, particularly the efficacy of block smoothing compared with simple pointwise smoothing, and illustrate the accuracy and stability of implicit–explicit time integration.

Key words. unsaturated poroelasticity, heterogeneous porous media, implicit–explicit time integration, additive scheme, multiscale Vanka smoother, local spectral coarse space, two-grid solver

AMS subject classifications. 65M55, 65M60, 65N22, 76S05

1. Introduction. Accurately simulating unsaturated flow is essential in hydrology, soil science, reservoir simulation, and environmental engineering. It describes water movement through porous media where both air and water coexist, and is governed by nonlinear partial differential equations (PDEs) combining Darcy's law and the continuity equation [12]. The model includes highly nonlinear relationships between moisture content, pressure, and hydraulic conductivity, often described by empirical functions such as the van Genuchten or Brooks–Corey models [7, 36]. Coupling mechanics with flow is crucial, as changes in water content affect both permeability and the mechanical properties of the porous matrix, leading to complex interactions [11, 34]. Numerical modeling of such coupled processes in heterogeneous media is challenging due to strong nonlinearity, multiphysics interactions, and scale dependencies. Advances in numerical methods can directly enhance the accuracy and efficiency of simulations in geotechnical engineering, reservoir modeling, and environmental hydrology [39].

Classical numerical schemes for solving time-dependent problems are based on explicit or implicit time approximation. Explicit methods are straightforward in implementation, but can require very small time steps for stability, making them impractical for problems with heterogeneous properties. Implicit schemes enhance stability for stiff equations, but require computationally expensive linear and nonlinear solves. An alternative approach involves

---

[†]Corresponding author. Department of Mathematics and Statistics, Texas A&M University - Corpus Christi, Corpus Christi, Texas, TX 78412, USA (maria.vasilyeva@tamucc.edu)

[‡]Theoretical Division, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, New Mexico, NM 87545, USA (southworth@lanl.gov)

[§]Department of Mathematics, University of Houston, Houston, Texas, TX 77204, USA ({yhe43, mwang55}@central.uh.edu)

combining the two methods using an implicit–explicit (ImEx) time-stepping scheme, which applies an explicit scheme to non-stiff components and an implicit scheme to stiff components that require improved stability [5, 35]. However, even in ImEx schemes, the bulk of the computational cost is in solving the implicit equations.

Recent studies have demonstrated the effectiveness of various multigrid methods on poroelastic models, e.g., [2, 25]. For challenging problems, robust multigrid methods can require more advanced smoothers than simple pointwise methods such as Jacobi or Gauss–Seidel. This has led to renewed interest in block- or patch-based smoothers, e.g., [14], along the lines of Schwarz or Vanka methods [37], and related to (spectral) domain decomposition methods [31]. Such methods can be overlapping or non-overlapping with respect to patches or "subdomains" that are locally solved, and additive or multiplicative in nature. Multicolor smoothers, e.g., [17, 41], that combine ideas from additive and multiplicative methods are often used for improved parallelization over multiplicative methods and improved performance over additive methods.

In this work, we develop both a time-discretization strategy and corresponding two-level solver framework for unsaturated flow in heterogeneous poroelastic media. Our main contributions are as follows:

1. We consider nonlinear poroelastic systems with high spatial variations in conductivity and elastic properties, as well as sharp wetting fronts, requiring fine-scale resolution.
2. We develop a *novel implicit–explicit null (EIN) scheme*, splitting the operator into a linear approximation fixed for all time (treated implicitly) and a nonlinear residual (treated explicitly). It is proven that the proposed splitting ensures unconditional linear stability, and the reuse of the linear implicit component across all time is key to constructing an efficient linear solver.
3. We propose a *new and efficient multiscale two-grid solver*, consisting of:
   - a coarse-grid solver based on *local spectral enrichment* for flow and mechanics, and
   - a *new coupled multiscale Vanka smoother* with overlapping subdomains defined by multiscale basis supports, coarse cells, or extended coarse-cell neighborhoods, and multicolor additive implementations.
4. We demonstrate that, despite the additional cost of computing multiscale basis functions and Vanka smoothers, these components can be precomputed *offline* and reused, yielding highly efficient simulations for nonlinear unsaturated poroelastic problems.

The paper is organized as follows. In Section 2, we describe the continuous PDE problem formulation for unsaturated flow in poroelastic media. Section 3 proceeds to introduce a finite-element approximation in space, and Section 4 proposes a new implicit–explicit time integration scheme. We then provide a stability analysis demonstrating that the proposed additive partitioning of the operator leads to an unconditionally stable scheme. In Section 5, we propose the construction of a multiscale two-grid solver based on a multiscale Vanka approach, and describe the construction of local spectral multiscale basis functions for the flow and mechanics parts of the system. The performance of our proposed two-grid solver with local smoothing iterations for the nonlinear poroelastic problem in heterogeneous media is demonstrated in Section 6. Conclusions are discussed in Section 7.

2. Problem formulation. We consider the unsaturated poroelasticity model that describes the coupled behavior of fluid flow and mechanical deformation in partially saturated porous media. Specifically, we utilize the model from [18, 38, 39] with an additional nonlinearity to account for elastic properties [24].

2.1. Unsaturated flow. The Richards equation given below describes an unsaturated flow in porous media, which combines Darcy's law for variably saturated flow with the mass conservation equation [12]:

$$
(2.1) \qquad c_w(x, h_w)\frac{\partial h_w}{\partial t} - \nabla \cdot (k_w(x, h_w)(\nabla h_w + \nabla \zeta)) = f_w.
$$

Here θ is the water content, h_w is the pressure head, ζ is the elevation head, f_w is a fluid source/sink term, k_w is the hydraulic conductivity, and $c_w(x, h_w) = \partial \theta / \partial h_w$.

The nonlinear conductivity coefficient and water content depend on the pressure head h_w. The shape of the water retention curves can be characterized by the van Genuchten model [36]:

$$
(2.2) \qquad \theta(h_w) = \theta_r + (\theta_s - \theta_r)[1 + (\beta|h_w|)^{n_\theta}]^{-m_\theta}, \quad h_w < 0.
$$

Here θ_s is the saturated water content, θ_r is the residual water content, β is an empirical parameter related to the inverse of the air entry pressure, and n_θ and m_θ are fitting parameters with $m_\theta = 1 - 1/n_\theta$. The unsaturated hydraulic conductivity function is defined as

$$
(2.3) \qquad k_w(x, h_w) = k_s(x)(S_e(h_w))^\eta \big\{1 - \big[1 - \big(S_e(h_w)\big)^{1/m_\theta}\big]^{m_\theta}\big\}^2,
$$

where k_s is the saturated hydraulic conductivity, $S_e(h_w) = (\theta(h_w) - \theta_r)/(\theta_s - \theta_r)$ is the effective saturation, and η is a pore connectivity parameter.

2.2. Mechanics. For the mechanics, we consider the following momentum balance for the porous medium's displacement under the quasi-static and small-deformations assumptions [18, 38, 39]

$$
-\nabla \cdot \sigma_T = \rho_b g,
$$

where

$$
\sigma_T = \sigma - \alpha S p \mathcal{I}, \quad \sigma(u) = 2\mu \varepsilon(u) + \lambda(\nabla \cdot u)\mathcal{I}, \quad \varepsilon(u) = 0.5(\nabla u + (\nabla u)^T).
$$

Here, σ_T is the total stress tensor, σ is the stress tensor, p is the water pressure, S is the water saturation, α is the Biot coefficient, u is the displacement vector, ε is the strain tensor, $\rho_b = \phi S \rho_w + (1 - \phi)\rho_s$ is the bulk density, g is the gravity vector, ϕ is the porosity, ρ_w is the water density, ρ_s is the solid-phase density, $\mathcal{I}$ is the identity tensor, and λ and μ are the Lamé coefficients.

Additionally, we consider nonlinear elastic properties that describe the effect of changes in water content on the mechanical properties [24]:

$$
\lambda(x, h_w) = \frac{E(x, h_w)\nu}{(1 + \nu)(1 - 2\nu)}, \quad \mu(x, h_w) = \frac{E(x, h_w)}{2(1 + \nu)}.
$$

Here ν is the Poisson's ratio and $E(x, h_w)$ is the Young's modulus given by

$$
(2.4) \quad E(x, h_w) = E_d(x) + (E_w(x) - E_d(x))(S_e(h_w))^\zeta = E_d(x)[1 + (r - 1)(S_e(h_w))^\zeta],
$$

where $r_E = E_w(x)/E_d(x)$. The elastic properties of the porous matrix are water-dependent, meaning that soils and rocks tend to soften as they absorb water, leading to increased deformations. Here E_d and E_w are the Young's modulus of the dry and wet states, respectively, and ζ is the empirical fitting parameter [24].

We note that the relationships (2.2) and (2.3) are expressed in terms of water content and pressure. Therefore, we adapt the original van Genuchten model to water pressure and saturation representation by using the following relations:

$$S = \theta/\phi \quad \text{and} \quad p = \rho_w g h_w.$$

Here, instead of formulation (2.1) for (θ, h_w), we write a mass conservation equation in terms of (S, p) [18, 38, 39]. A full derivation from physical principles can be found in Appendix A.

2.3. Problem setting and assumptions. We let $\Omega \subset \mathbb{R}^d$ be the computational domain and consider the following coupled nonlinear poroelasticity system in Ω:

(2.5)
$$c(x,p)\frac{\partial p}{\partial t} + \alpha S(p)\frac{\partial(\nabla \cdot u)}{\partial t} - \nabla \cdot (\kappa(x,p)\nabla p) = \tilde{f}, \quad x \in \Omega, \quad t > 0,$$
$$-\nabla \cdot \sigma(u) + \alpha\nabla(S(p)p) = \rho_b g, \quad x \in \Omega,$$

with $\tilde{f} = f + \nabla \cdot (\kappa(x,p)\nabla(\rho_w g))$ and

$$c(x,p) = [\phi C_w + (\alpha - \phi)C_s S]S + [\phi + (\alpha - \phi)C_s Sp]S', \quad \kappa(x,p) = k_s(x)\frac{k_{rw}(S)}{\mu_w}.$$

Here k is the intrinsic permeability tensor, μ_w is the water dynamic viscosity, k_{rw} is the relative permeability, C_w is the water compressibility, C_s is the compressibility of the solid grains, $S = S(p)$, and $S' = \partial S/\partial p$.

We supplement equations (2.5) with the following initial conditions,

(2.6)
$$p = p^0, \quad u = u^0, \quad x \in \Omega, \quad t = 0,$$

and boundary conditions,

(2.7)
$$-\kappa\nabla p \cdot n = \gamma(p - p_1), \quad x \in \Gamma_p, \qquad -\kappa\nabla p \cdot n = 0, \quad x \in \partial\Omega/\Gamma_p, \quad t > 0,$$
$$u = 0, \qquad x \in \Gamma_u, \qquad \sigma \cdot n = 0, \quad x \in \partial\Omega/\Gamma_u, \quad t > 0.$$

Here p_1 is the given pressure on the boundary Γ_p, and γ is the exchange coefficient in a Robin-type boundary condition.

We assume the following:
- $k_s(x)$ and $E_d(x)$ are the intrinsic permeability tensor and drained Young's modulus, respectively, which may vary spatially.
- μ_w is the constant water dynamic viscosity, $k_{rw}(S)$ is the relative permeability, and $S(p)$ is the saturation function, modeled using the nonlinear van Genuchten relation.
- C_w is the constant water compressibility, and C_s is the constant solid-grain compressibility.
- ϕ is the constant porosity, α is the constant Biot coefficient, ρ_b is the constant bulk density, and g is the gravity vector.

For the analysis and numerical solution, we assume that the following hold:
- $c(x,p) > 0$ and $\kappa(x,p)$ is symmetric positive definite for all admissible x and p.
- $S(p)$ is smooth and monotone increasing, with $0 \le S(p) \le 1$.
- $k_s(x)$ and $E_d(x)$ are bounded and strictly positive.
- The domain Ω is Lipschitz, and the boundary portions Γ_p and Γ_u have positive measure.

Discretizations in space and time are detailed next.

3. Spatial discretization and semi-implicit time integration. Here we will detail a finite-element method (FEM) discretization with linear basis functions for both the pressure and displacement equations. In principle, an additional regularization can be included in the pressure equation to prevent oscillations and stabilize the discretization for certain parameter regimes [1, 19, 20, 22, 28], but here we focus on the standard case without additional regularization.

Let

$$V = \{v \in [H^1(\Omega)]^d : v = 0 \text{ on } \Gamma_u\}, \quad Q = H^1(\Omega).$$

We define a variational formulation of the unsaturated poroelasticity problem (2.5)–(2.7) as follows: Find $(p, u) \in Q \times V$ such that

$$\int_\Omega \alpha S(p) \frac{\partial (\nabla \cdot u)}{\partial t} r \, dx + \int_\Omega c(x, p) \frac{\partial p}{\partial t} r \, dx + \int_\Omega \kappa(x, p) \nabla p \cdot \nabla r \, dx$$
$$+ \int_{\Gamma_p} \gamma p r \, ds = \int_\Omega \tilde{f} r \, dx + \int_{\Gamma_p} \gamma p_1 r \, ds, \quad \forall \, r \in Q,$$

$$\int_\Omega \sigma(u) : \varepsilon(v) \, dx + \int_\Omega \alpha \nabla (S(p) p) \cdot v \, dx = \int_\Omega \rho_b g v \, dx, \quad \forall \, v \in V.$$

We use a finite-element method for spatial discretization. Let $\mathcal{T}^h$ be a partition of the domain Ω, $\mathcal{E}^b$ be a set of all boundary interfaces, and $\mathcal{E}^b_p$ be a subset of the boundary interfaces $\mathcal{E}^b_p = \mathcal{E}^b \cap \Gamma_p$. Consider the first-order backward Euler implicit scheme with constant time-step size τ. Then we obtain the following discrete system in matrix form for $(p^{n+1}_h, u^{n+1}_h) \in Q_h \times V_h$:

$$(3.1) \quad \frac{1}{\tau} \begin{bmatrix} M^{n+1}_h & \alpha D^{n+1}_h \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p^{n+1}_h - p^n_h \\ u^{n+1}_h - u^n_h \end{bmatrix} + \begin{bmatrix} A^{n+1}_h & 0 \\ \alpha G^{n+1}_h & K^{n+1}_h \end{bmatrix} \begin{bmatrix} p^{n+1}_h \\ u^{n+1}_h \end{bmatrix} = \begin{bmatrix} F^{n+1}_{p,h} \\ F^{n+1}_{u,h} \end{bmatrix},$$

where $c^{n+1} = c(p^{n+1})$, $k^{n+1} = k(x, p^{n+1})$, and

$$A^{n+1}_h = \left[a^{n+1}_{ij} = \int_\Omega \kappa^{n+1} \nabla \phi_i \cdot \nabla \phi_j \, dx \right],$$

$$K^{n+1}_h = \left[b^{n+1}_{ln} = \int_\Omega [2\mu^{n+1} \varepsilon(\Phi_l) : \varepsilon(\Phi_n) \, dx + \lambda^{n+1} \nabla \cdot \Phi_l \, \nabla \cdot \Phi_n] \, dx \right],$$

$$D^{n+1}_h = \left[d^{n+1}_{il} = \int_\Omega S(p^{n+1}) \phi_i \nabla \cdot \Phi_l \, dx \right],$$

$$G^{n+1}_h = \left[g^{n+1}_{li} = \int_\Omega \nabla (S(p^{n+1}) \phi_i) \cdot \Phi_l \, dx \right],$$

$$F^{n+1}_{p,h} = \left[f^{n+1}_{p,i} = \int_\Omega \tilde{f}^{n+1} \phi_i \, dx + \int_{\Gamma_p} \gamma p_1 \phi_i \, ds \right],$$

$$F^{n+1}_{u,h} = \left[f^{n+1}_{u,l} = \int_\Omega \rho^{n+1}_b g \, \Phi_l \, dx \right],$$

$$M^{n+1}_h = \left[m^{n+1}_{ij} = \int_\Omega c^{n+1} \phi_i \phi_j \, dx \right],$$

with

$$u^{n+1}_h = \sum_{l=1}^{N_v} u^{n+1}_l \Phi_l \quad \text{and} \quad p^{n+1}_h = \sum_{i=1}^{N_v} p^{n+1}_i \phi_i.$$

Here $\{\Phi_l\}$ are the linear basis functions for displacements, $\{\phi_i\}$ are the linear basis functions for pressure, and N_v is the number of vertices on the grid $\mathcal{T}_h$.

Each implicit time step requires solving a nonlinear system. To linearize, we will use a Picard iteration in the form

$$(3.2) \qquad \begin{aligned} &\frac{1}{\tau} \begin{bmatrix} M_h^{n+1,(m)} & \alpha D_h^{n+1,(m)} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_h^{n+1,(m+1)} - p_h^n \\ u_h^{n+1,(m+1)} - u_h^n \end{bmatrix} \\ &+ \begin{bmatrix} A_h^{n+1,(m)} & 0 \\ \alpha G_h^{n+1,(m)} & K_h^{n+1,(m)} \end{bmatrix} \begin{bmatrix} p_h^{n+1,(m+1)} \\ u_h^{n+1,(m+1)} \end{bmatrix} = \begin{bmatrix} F_{p,h}^{n+1,(m)} \\ F_{u,h}^{n+1,(m)} \end{bmatrix}, \end{aligned}$$

where (m) denotes nonlinear iteration, and the process continues until a desired tolerance between successive iterates is obtained, $\|p_h^{n+1,(m+1)} - p_h^{n+1,(m)}\| \le \varkappa_p$ and $\|u_h^{n+1,(m+1)} - u_h^{n+1,(m)}\| \le \varkappa_u$, for tolerances $\varkappa_p$ and $\varkappa_u$.

Using an analogous linearization, we may also linearize (2.7) on the level of the time integration scheme using the solution from the previous time step to evaluate certain nonlinear quantities:

$$(3.3) \qquad \frac{1}{\tau} \begin{bmatrix} M_h^n & \alpha D_h^n \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_h^{n+1} - p_h^n \\ u_h^{n+1} - u_h^n \end{bmatrix} + \begin{bmatrix} A_h^n & 0 \\ \alpha G_h^n & K_h^n \end{bmatrix} \begin{bmatrix} p_h^{n+1} \\ u_h^{n+1} \end{bmatrix} = \begin{bmatrix} F_{p,h}^n \\ F_{u,h}^n \end{bmatrix}.$$

Such an approach is a simple way to reduce computational costs by solving only a single linear system each time step, rather than resolving a fully nonlinear system, albeit typically at a cost of reduced stability and/or larger error constant. This approach balances stability, accuracy, and computational efficiency for the nonlinear poroelastic system, and belongs to the broader class of first-order semi-implicit integrators and nonlinear operator partitions [10].

4. Implicit–explicit scheme for time approximation. To approximate the nonlinear flow problem in time, we now construct an additive EIN splitting, with an implicit linear term that is fixed for all time. We follow our previous work [20], in which we developed a splitting scheme for the poroelasticity equations to construct stable solvers and decouple the displacement and pressure equations. While the present approach is still based on additive splitting, it is conceptually different. Here, we propose using additive splitting to derive an EIN scheme.

4.1. Additive scheme. In the poroelasticity system (2.5), the second equation does not contain a time derivative, and the system is in the form of differential algebraic equations. To construct an additive scheme and analyze the stability of the resulting linearized formulation, we begin by differentiating the displacement equation with respect to time:

$$c(x,p)\frac{\partial p}{\partial t} + \alpha S(p)\frac{\partial(\nabla \cdot u)}{\partial t} - \nabla \cdot (\kappa(x,p)\nabla p) = \tilde{f}, \qquad x \in \Omega, \quad t > 0,$$

$$-\frac{\partial(\nabla \cdot \sigma(u))}{\partial t} + \alpha\frac{\partial(S(p)\nabla p)}{\partial t} = \frac{\partial(\rho_b g)}{\partial t}, \quad x \in \Omega, \quad t > 0,$$

with initial conditions $p = p^0$ and $u = u^0$ such that

$$-\nabla \cdot \sigma(u^0) + \alpha\nabla p^0 = \rho_b g, \quad x \in \Omega, \quad t = 0.$$

Then, for a semi-discrete system, we have the following matrix form for $U = (p_h, u_h) \in Q_h \times V_h$:

$$(4.1) \qquad\qquad\qquad \mathbb{C}\frac{\partial U}{\partial t} + \mathbb{L}U = F,$$

with

$$\mathbb{C} = \begin{bmatrix} M_h & \alpha D_h \\ \alpha G_h & K_h \end{bmatrix}, \quad \mathbb{L} = \begin{bmatrix} A_h & 0 \\ 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} F_{p,h} \\ \delta F_{u,h} \end{bmatrix}.$$

The resulting semi-discrete system can be viewed as an initial-value problem for a system of ordinary differential equations [20, 27].

The standard implicit approximation for the problem (4.1) has the following form for $U^{n+1} = (p_h^{n+1}, u_h^{n+1})$:

$$\mathbb{C}^n \frac{U^{n+1} - U^n}{\tau} + \mathbb{L}^n U^{n+1} = F^n,$$

with

$$\mathbb{C}^n = \begin{bmatrix} M_h^{n+1} & \alpha D_h^{n+1} \\ \alpha G_h^{n+1} & K_h^{n+1} \end{bmatrix} \quad \text{and} \quad \mathbb{L}^n = \begin{bmatrix} A_h^{n+1} & 0 \\ 0 & 0 \end{bmatrix}.$$

To eliminate the need for reassembling the problem operator at each time step, we define the following additive representation of the problem operators:

$$\mathbb{C}^n = \mathbb{C}^{(lin)} + \mathbb{C}^{n,(nl)} = \begin{bmatrix} M_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix} + \begin{bmatrix} M_h^{n,(nl)} & \alpha D_h^{n,(nl)} \\ \alpha G_h^{n,(nl)} & K_h^{n,(nl)} \end{bmatrix},$$

$$\mathbb{L}^n = \mathbb{L}^{(lin)} + \mathbb{L}^{n,(nl)} = \begin{bmatrix} A_h^{(lin)} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A_h^{n,(nl)} & 0 \\ 0 & 0 \end{bmatrix}.$$

Here $M^{(lin)}$, $D^{(lin)}$, $G^{(lin)}$, $A^{(lin)}$, and $K^{(lin)}$ are linear operators, and $M^{n,(nl)}$, $D^{n,(nl)}$, $G^{n,(nl)}$, $A^{n,(nl)}$, and $K^{n,(nl)}$ are nonlinear residual operators, which depend on current solution (p_h^n, u_h^n) and should be updated at each time step, i.e.,

$$M_h^{n,(nl)} = M_h^n - M_h^{(lin)}, \quad D_h^{n,(nl)} = D_h^n - D_h^{(lin)}, \quad G_h^{n,(nl)} = G_h^n - G_h^{(lin)},$$
$$A_h^{n,(nl)} = A_h^n - A_h^{(lin)}, \quad K_h^{n,(nl)} = K_h^n - K_h^{(lin)}.$$

Such an additive representation separates the nonlinear parts, and by approximating it explicitly, we obtain a linear system without updating the operator at each time layer.

Then, the proposed implicit–explicit scheme can be written as a three-step scheme:

$$\underbrace{\mathbb{C}^{(lin)} \frac{U^{n+1} - U^n}{\tau} + \mathbb{L}^{(lin)} U^{n+1}}_{\text{linear/implicit}} + \underbrace{\mathbb{C}^{n,(nl)} \frac{U^n - U^{n-1}}{\tau} + \mathbb{L}^{n,(nl)} U^n}_{\text{nonlinear/explicit}} = F^n,$$

or

$$\underbrace{\begin{bmatrix} M_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix} \begin{bmatrix} p_h^{n+1} - p_h^n \\ u_h^{n+1} - u_h^n \end{bmatrix} + \begin{bmatrix} \tau A_h^{(lin)} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_h^{n+1} \\ u_h^{n+1} \end{bmatrix}}_{\text{linear/implicit}}$$

(4.2)

$$+ \underbrace{\begin{bmatrix} M_h^{n,(nl)} & \alpha D_h^{n,(nl)} \\ \alpha G_h^{n,(nl)} & K_h^{n,(nl)} \end{bmatrix} \begin{bmatrix} p_h^n - p_h^{n-1} \\ u_h^n - u_h^{n-1} \end{bmatrix} + \begin{bmatrix} \tau A_h^{n,(nl)} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_h^n \\ u_h^n \end{bmatrix}}_{\text{nonlinear/explicit}}$$

$$= \begin{bmatrix} \tau F_{p,h}^n \\ F_{u,h}^n - F_{u,h}^{n-1} \end{bmatrix}.$$

The main question in this implicit–explicit approximation is how to split the operators to ensure the unconditional stability of the scheme. We will discuss this in the following sections.

4.2. Stability analysis. Traditionally, splitting schemes for the poroelasticity system have been developed and analyzed by separating the displacement and pressure calculations, allowing each variable to be solved independently. Common approaches include the undrained split, fixed-stress split, drained split, and fixed-strain split methods, whose stability properties have been studied extensively in the literature [19, 20, 26]. Our proposed approach differs from traditional splitting methods for poroelasticity and is specifically designed to address nonlinearities, where we treat part of the coupling implicitly to avoid these stability problems.

We use a linear stability analysis of time-dependent PDEs and derive the unconditional stability of the proposed implicit–explicit scheme [35]. The stability here does not imply nonlinear stability and should be interpreted within the framework of linearized analysis. Next, we show that the scheme is unconditionally stable provided that the explicitly treated residual is bounded by the implicitly treated approximation [4, 9, 21]. For notation, we let $A_h^{(nl)} \leq (1 - \varrho)A_h^{n,(lin)}$ imply that, for $A_h^{(nl)}$ linearized about any state, the operator $(1 - \varrho)A_h^{n,(lin)} - A_h^{(nl)}$ is symmetric positive definite (SPD).

THEOREM 4.1. *Let nonlinear discretization operators be linearized about some valid state, and assume all linearized discretization operators and corresponding linear approximations, e.g., $A_h^{(nl)}$ and $A_h^{n,(lin)}$, are symmetric. Assume that*

$$A_h^{(nl)} \leq (1 - \varrho)A_h^{n,(lin)}, \quad K_h^{(nl)} \leq (1 - \varrho)K_h^{n,(lin)}, \quad M_h^{(nl)} \leq (1 - \varrho)M_h^{n,(lin)},$$
$$D_h^{(nl)} \leq (1 - \varrho)D_h^{n,(lin)}, \quad G_h^{(nl)} \leq (1 - \varrho)G_h^{n,(lin)},$$

with $0 < \varrho < 1$. Then the solution of the discrete problem (4.2) *is stable.*

Proof. We use a general framework for the three-step scheme (4.2) and represent blocks of the system as follows [35]:

$$M_h^{(lin)}\frac{p_h^{n+1} - p_h^n}{\tau} + M_h^{n,(nl)}\frac{p_h^n - p_h^{n-1}}{\tau} + A_h^{(lin)}p_h^{n+1} + A_h^{n,(nl)}p_h^n$$

$$= B_{11}\frac{p_h^{n+1} - p_h^{n-1}}{2\tau} + R_{11}(p_h^{n+1} - 2p_h^n + p_h^{n-1}) + A_{11}p_h^n,$$

$$K_h^{(lin)}\frac{u_h^{n+1} - u_h^n}{\tau} + K_h^{n,(nl)}\frac{u_h^n - u_h^{n-1}}{\tau} = B_{22}\frac{u_h^{n+1} - u_h^{n-1}}{2\tau} + R_{22}(u_h^{n+1} - 2u_h^n + u_h^{n-1}),$$

$$D_h^{(lin)}\frac{u_h^{n+1} - u_h^n}{\tau} + D_h^{n,(nl)}\frac{u_h^n - u_h^{n-1}}{\tau} = B_{12}\frac{u_h^{n+1} - u_h^{n-1}}{2\tau} + R_{12}(u_h^{n+1} - 2u_h^n + u_h^{n-1}),$$

$$G_h^{(lin)}\frac{p_h^{n+1} - p_h^n}{\tau} + G_h^{n,(nl)}\frac{p_h^n - p_h^{n-1}}{\tau} = B_{21}\frac{p_h^{n+1} - p_h^{n-1}}{2\tau} + R_{21}(p_h^{n+1} - 2p_h^n + p_h^{n-1})$$

with

$$B_{11} = M_h^{(lin)} + M_h^{n,(nl)} + \tau A_h^{(lin)}, \quad R_{11} = \frac{1}{2\tau}(M_h^{(lin)} - M_h^{n,(nl)} + \tau A_h^{(lin)}),$$

$$A_{11} = A_h^{(lin)} + A_h^{n,(nl)} = A_h^n,$$

$$B_{22} = K_h^{(lin)} + K_h^{n,(nl)}, \quad R_{22} = \frac{1}{2\tau}(K_h^{(lin)} - K_h^{n,(nl)}),$$

$$B_{12} = \alpha\big(D_h^{(lin)} + D_h^{n,(nl)}\big), \quad R_{12} = \frac{\alpha}{2\tau}(D_h^{(lin)} - D_h^{n,(nl)}),$$

$$B_{21} = \alpha\big(G_h^{(lin)} + G_h^{n,(nl)}\big), \quad R_{21} = \frac{\alpha}{2\tau}(G_h^{(lin)} - G_h^{n,(nl)}).$$

We let $y_1^n = p_h^n$ and $y_2^n = u_h^n$, and introduce auxiliary variables in order to represent the scheme (4.2) as a two-step scheme:

$$w_i^n = y_i^n - y_i^{n-1}, \quad v_i^n = \frac{y_i^n + y_i^{n-1}}{2}, \quad i = 1, 2.$$

For w_i^n and v_i^n, we have

$$w_i^{n+1} + w_i^n = 2(v_i^{n+1} - v_i^n) = y_i^{n+1} - y_i^{n-1},$$
$$w_i^{n+1} - w_i^n = y_i^{n+1} - 2y_i^n + y_i^{n-1},$$
$$v_i^{n+1} + v_i^n = 2v_i^n + (v_i^{n+1} - v_i^n),$$
$$y_i^n = \tfrac{1}{2}(v_i^{n+1} + v_i^n) - \tfrac{1}{4}(w_i^{n+1} - w_i^n).$$

It follows that

$$B_{ij}\frac{y_j^{n+1} - y_j^{n-1}}{2\tau} + R_{ij}(y_j^{n+1} - 2y_j^n + y_j^{n-1}) = B_{ij}\frac{w_j^{n+1} + w_j^n}{2\tau} + R_{ij}(w_j^{n+1} - w_j^n)$$

and

$$B_{11}\frac{y_1^{n+1} - y_1^{n-1}}{2\tau} + R_{11}(y_1^{n+1} - 2y_1^n + y_1^{n-1}) + A_{11}y_1^n$$
$$= B_{11}\frac{w_1^{n+1} + w_1^n}{2\tau} + R_{11}(w_1^{n+1} - w_1^n) + \frac{1}{2}A_{11}(v_1^{n+1} + v_1^n) - \frac{1}{4}A_{11}(w_1^{n+1} - w_1^n)$$
$$= B_{11}\frac{w_1^{n+1} + w_1^n}{2\tau} + \left(R_{11} - \frac{1}{4}A_{11}\right)(w_1^{n+1} - w_1^n) + \frac{1}{2}A_{11}(v_1^{n+1} + v_1^n)$$
$$= \frac{1}{\tau}B_{11}(v_1^{n+1} - v_1^n) + \left(R_{11} - \frac{1}{4}A_{11}\right)(w_1^{n+1} - w_1^n) + \frac{1}{2}A_{11}(v_1^{n+1} - v_1^n) + A_{11}v_1^n$$
$$= \frac{1}{\tau}\left(B_{11} + \frac{\tau}{2}A_{11}\right)(v_1^{n+1} - v_1^n) + Q_{11}(w_1^{n+1} - w_1^n) + A_{11}v_1^n,$$

with

$$Q_{11} = \left(R_{11} - \frac{1}{4}A_{11}\right) = \frac{1}{2\tau}(M_h^{(lin)} - M_h^{n,(nl)}) + \frac{1}{4}(A_h^{(lin)} - A_h^{n,(nl)}).$$

Then, we can rewrite (4.2) as follows:

(4.3)
$$\left(B_{11} + \frac{\tau}{2}A_{11}\right)\frac{v_1^{n+1} - v_1^n}{\tau} + Q_{11}(w_1^{n+1} - w_1^n) + A_{11}v_1^n$$
$$+ B_{12}\frac{v_2^{n+1} - v_2^n}{\tau} + R_{12}(w_2^{n+1} - w_2^n) = F_{p,h}^n,$$
$$B_{21}\frac{v_1^{n+1} - v_1^n}{\tau} + R_{21}(w_1^{n+1} - w_1^n)$$
$$+ B_{22}\frac{v_2^{n+1} - v_2^n}{\tau} + R_{22}(w_2^{n+1} - w_2^n) = F_{u,h}^n - F_{u,h}^{n-1}.$$

Additionally, we add the equations for w_1 and w_2 to produce a general system with square matrices [35]:

$$-\tau Q_{ij}\frac{v_j^{n+1} - v_j^n}{\tau} + \frac{\tau}{2}Q_{ij}\frac{w_j^{n+1} - w_j^n}{\tau} + Q_{ij}w_i^n$$
$$= -Q_{ij}(v_j^{n+1} - v_j^n) + \frac{1}{2}Q_{ij}(w_j^{n+1} + w_j^n) = 0,$$

where the matrices Q_{ij} will be defined later in order to ensure that the resulting general block matrix is skew-symmetric.

We let $Y = (v_1, w_1, v_2, w_2)$ and rewrite the system (4.3) in the following form:

$$(4.4) \qquad \mathbb{B}\frac{Y^{n+1} - Y^n}{\tau} + \mathbb{A}Y^n = F,$$

with

$$\mathbb{B} = \begin{bmatrix} B_{11} + \frac{1}{2}\tau A_{11} & \tau Q_{11} & B_{12} & \tau R_{12} \\ -\tau Q_{11} & \frac{1}{2}\tau Q_{11} & \tau Q_{12} & -\frac{1}{2}\tau Q_{12} \\ B_{21} & \tau R_{21} & B_{22} & \tau R_{22} \\ \tau Q_{21} & -\frac{1}{2}\tau Q_{21} & -\tau Q_{22} & \frac{1}{2}\tau Q_{22} \end{bmatrix},$$

$$\mathbb{A} = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & Q_{11} & 0 & -Q_{12} \\ 0 & 0 & 0 & 0 \\ 0 & -Q_{21} & 0 & Q_{22} \end{bmatrix}, \quad \text{and} \quad F^n = \begin{bmatrix} F_1^n \\ 0 \\ F_2^n \\ 0 \end{bmatrix},$$

where $F_1^n = F_{p,h}^n$ and $F_2^n = F_{u,h}^n - F_{u,h}^{n-1}$. Here, we set $Q_{22} = R_{22}$, $Q_{12} = R_{12}$, and $Q_{21} = R_{21}$ to preserve the skew-symmetry of the block matrices $\mathbb{B}$ and $\mathbb{A}$ with $R_{12} = -R_{21}^T$.

Next, we employ the additive representation

$$\mathbb{B} = \mathbb{B}^{(0)} + \mathbb{B}^{(1)} \quad \text{and} \quad \mathbb{A} = \mathbb{A}^{(0)} + \mathbb{A}^{(1)},$$

where the superscripts (0) and (1) correspond to the linear and nonlinear parts, respectively, so that

$$\mathbb{W} = \left(\mathbb{B} - \frac{\tau}{2}\mathbb{A}\right) = \begin{bmatrix} B_{11} & \tau Q_{11} & B_{12} & \tau Q_{12} \\ -\tau Q_{11} & 0 & \tau Q_{12} & 0 \\ B_{21} & \tau Q_{21} & B_{22} & \tau Q_{22} \\ \tau Q_{21} & 0 & -\tau Q_{22} & 0 \end{bmatrix} = \mathbb{W}^{(0)} + \mathbb{W}^{(1)}.$$

With $A_h = A_h^T$, $M_h = M_h^T$, $K_h = K_h^T$, $D_h = -G_h^T$, and $\check{Y} = (\check{v}_1, \check{w}_1, \check{v}_2, \check{w}_2)$, we have

$$(\mathbb{A}Y, \check{Y}) = (A_{11}v_1, \check{v}_1) + (Q_{11}w_1, \check{w}_1) - (Q_{12}w_2, \check{w}_1) - (Q_{21}w_1, \check{w}_2) + (Q_{22}w_2, \check{w}_2)$$
$$= (A_{11}v_1, \check{v}_1) + (Q_{11}w_1, \check{w}_1) + (Q_{22}w_2, \check{w}_2) = (\mathbb{A}\check{Y}, Y),$$

$$(\mathbb{W}Y, Y) = (B_{11}v_1, v_1) + \tau(Q_{11}w_1, v_1) + (B_{12}v_2, v_1)$$
$$+ \tau(Q_{12}w_2, v_1) - \tau(Q_{11}v_1, w_1) + \tau(Q_{12}v_2, w_1)$$
$$+ (B_{21}v_1, v_2) + \tau(Q_{21}w_1, v_2) + (B_{22}v_2, v_2)$$
$$+ \tau(Q_{22}w_2, , v_2) + \tau(Q_{21}v_1, w_2) - \tau(Q_{22}v_2, w_2)$$
$$= (B_{11}v_1, v_1) + (B_{22}v_2, v_2) > 0.$$

Furthermore, $\mathbb{W}$ is skew-symmetric and we choose an additive representation that gives skew-symmetric matrices $\mathbb{W}^{(0)} > 0$ and $\mathbb{W}^{(1)} > 0$, where

$$\mathbb{W}^{(0)} = \begin{bmatrix} M_h^{(lin)} + \tau A_h^{(lin)} & \frac{1}{2}M_h^{(lin)} + \frac{1}{4}\tau A_h^{(lin)} & \alpha D_h^{(lin)} & \frac{1}{2}\alpha D_h^{(lin)} \\ -(\frac{1}{2}M_h^{(lin)} + \frac{1}{4}\tau A_h^{(lin)}) & 0 & \frac{1}{2}\alpha D_h^{(lin)} & 0 \\ \alpha G_h^{(lin)} & \frac{1}{2}\alpha G_h^{(lin)} & K_h^{(lin)} & \frac{1}{2}K_h^{(lin)} \\ \frac{1}{2}\alpha G_h^{(lin)} & 0 & -\frac{1}{2}K_h^{(lin)} & 0 \end{bmatrix},$$

$$\mathbb{W}^{(1)} = \begin{bmatrix} M_h^{n,(nl)} & -(\frac{1}{2}M_h^{n,(nl)} + \frac{1}{4}\tau A_h^{n,(nl)}) & \alpha D_h^{n,(nl)} & -\frac{1}{2}\alpha D_h^{n,(nl)} \\ (\frac{1}{2}M_h^{n,(nl)} + \frac{1}{4}\tau A_h^{n,(nl)}) & 0 & -\frac{1}{2}\alpha D_h^{n,(nl)} & 0 \\ \alpha G_h^{n,(nl)} & -\frac{1}{2}\alpha G_h^{n,(nl)} & K_h^{n,(nl)} & -\frac{1}{2}K_h^{n,(nl)} \\ -\frac{1}{2}\alpha G_h^{n,(nl)} & 0 & \frac{1}{2}K_h^{n,(nl)} & 0 \end{bmatrix}.$$

Then, we have

$$\mathbb{B}^{(0)}\frac{Y^{n+1} - Y^n}{\tau} + \mathbb{A}^{(0)}Y^n = \tilde{F}, \quad \tilde{F} = F - \mathbb{B}^{(1)}\frac{Y^{n+1} - Y^n}{\tau} - \mathbb{A}^{(1)}Y^n,$$

or

$$\mathbb{B}^{(0)}\frac{Y^{n+1} - Y^n}{\tau} + \mathbb{A}^{(0)}Y^n = \mathbb{B}^{(0)}\frac{Y^{n+1} - Y^n}{\tau} + \frac{1}{2}\mathbb{A}^{(0)}((Y^{n+1} + Y^n) - (Y^{n+1} - Y^n))$$

$$= \mathbb{W}^{(0)}\frac{Y^{n+1} - Y^n}{\tau} + \frac{1}{2}\mathbb{A}^{(0)}(Y^{n+1} + Y^n) = \tilde{F},$$

with

$$\tilde{F} = F - \mathbb{W}^{(1)}\frac{Y^{n+1} - Y^n}{\tau} - \frac{1}{2}\mathbb{A}^{(1)}(Y^{n+1} + Y^n).$$

Finally, by multiplying by $v = (Y^{n+1} - Y^n)/\tau$ and using the Cauchy–Schwarz and Young's inequalities, and assuming that

$$\mathbb{A}^{(1)} \leq (1 - \varrho)\mathbb{A}^{(0)}, \quad \mathbb{B}^{(1)} \leq (1 - \varrho)\mathbb{B}^{(0)}, \quad 0 < \varrho < 1,$$

we obtain

$$\left(\mathbb{W}^{(0)}\frac{Y^{n+1} - Y^n}{\tau}, \frac{Y^{n+1} - Y^n}{\tau}\right) + \frac{1}{2}\left(\mathbb{A}^{(0)}(Y^{n+1} + Y^n), \frac{Y^{n+1} - Y^n}{\tau}\right)$$

$$= \left(\tilde{F}, \frac{Y^{n+1} - Y^n}{\tau}\right)$$

$$\leq 2\epsilon\left\|\frac{Y^{n+1} - Y^n}{\tau}\right\|_{\mathbb{W}^{(0)}}^2 + \frac{1}{2\epsilon}\|F\|_{(\mathbb{W}^{(0)})^{-1}}^2$$

$$+ (1 - \varrho)\left(\mathbb{W}^{(0)}\frac{Y^{n+1} - Y^n}{\tau}, \frac{Y^{n+1} - Y^n}{\tau}\right)$$

$$+ (1 - \varrho)\frac{1}{2}\left(\mathbb{A}^{(0)}(Y^{n+1} + Y^n), \frac{Y^{n+1} - Y^n}{\tau}\right).$$

Then, with $(\mathbb{A}^{(0)}(Y^{n+1} + Y^n), Y^{n+1} - Y^n) = (\mathbb{A}^{(0)}Y^{n+1}, Y^{n+1}) - (\mathbb{A}^{(0)}Y^n, Y^n)$, we obtain

$$\frac{\varrho}{\tau^2}\|Y^{n+1} - Y^n\|_{\mathbb{W}^{(0)}}^2 + \frac{\varrho\tau}{2\tau^2}\|Y^{n+1}\|_{\mathbb{A}^{(0)}}^2$$

$$\leq \frac{\varrho\tau}{2\tau^2}\|Y^n\|_{\mathbb{A}^{(0)}}^2 + \frac{2\epsilon}{\tau^2}\|Y^{n+1} - Y^n\|_{\mathbb{W}^{(0)}}^2 + \frac{1}{2\epsilon}\|F\|_{(\mathbb{W}^{(0)})^{-1}}^2.$$

We set $\varepsilon = \varrho/2$ and get

$$\|Y^{n+1}\|^2_{\mathbb{A}^{(0)}} \leq \|Y^n\|^2_{\mathbb{A}^{(0)}} + \frac{2\tau}{\varrho^2}\|F\|^2_{(\mathbb{W}^{(0)})^{-1}}.$$

Therefore, we have the stability of the scheme (4.4) in $\mathbb{A}^{(0)}$ norm.   □

REMARK 4.2. We emphasize that the above result concerns the *linear stability* of the proposed implicit–explicit scheme. Nonlinear stability and convergence are not directly guaranteed by this analysis. However, in practice, the linear stability result is expected to provide useful guidance in regimes where the nonlinear dynamics are well resolved by the discrete time steps, and the linearized operators are bounded as assumed in the statement of Theorem 4.1. The numerical experiments in Section 6 confirm that the scheme remains robust and accurate for the considered range of time steps and problem parameters, suggesting that the linear stability analysis offers practical insight into the method's behavior. A detailed investigation of nonlinear stability and convergence will be pursued in future work.

4.3. Choice of the additive representation and system of linear equations. Let

$$\bar{c}(x) = \max_{p \in [p_{\min}, p_{\max}]} c(x, p), \quad \bar{S}(x) = \max_{p \in [p_{\min}, p_{\max}]} S(x, p),$$

$$\bar{\kappa}(x) = \max_{p \in [p_{\min}, p_{\max}]} \kappa(x, p), \quad \bar{E}(x) = \max_{p \in [p_{\min}, p_{\max}]} E(x, p).$$

Here, for example, $p_{\min}$ can be given as the boundary condition $p_1 = p_{\min}$, and $p_{\max}$ can be given as the initial condition $p_0 = p_{\max}$.

Then, we set

$$\bar{\sigma} = 2\bar{\mu}(x)\varepsilon(u) + \bar{\lambda}(x)(\nabla \cdot u)\mathcal{I}, \quad \bar{\lambda}(x) = \frac{\bar{E}(x)\nu}{(1+\nu)(1-2\nu)}, \quad \bar{\mu}(x) = \frac{\bar{E}(x)}{2(1+\nu)},$$

and

$$\tilde{c}(x, p) = c(x, p) - \bar{c}(x), \quad \tilde{\kappa}(x, p) = \kappa(x, p) - \bar{\kappa}(x),$$
$$\tilde{S}(x, p) = S(x, p) - \bar{S}(x), \quad \tilde{\mu}(x, p) = \mu(x, p) - \bar{\mu}(x), \quad \tilde{\lambda}(x, p) = \lambda(x, p) - \bar{\lambda}(x).$$

Finally, we have the following operators in the implicit–explicit scheme (4.2):

- *Linear/implicit*

$$A_h^{(lin)} = \left[a_{ij}^{(lin)} = \int_\Omega \bar{\kappa}\nabla\phi_i \cdot \nabla\phi_j \, dx\right],$$

$$M_h^{(lin)} = \left[m_{ij}^{(lin)} = \int_\Omega \bar{c}\phi_i\phi_j \, dx\right],$$

$$K_h^{(lin)} = \left[b_{ln}^{(lin)} = \int_\Omega [2\bar{\mu}\varepsilon(\Phi_l) : \varepsilon(\Phi_n) \, dx + \bar{\lambda}\nabla \cdot \Phi_l \, \nabla \cdot \Phi_n] \, dx\right],$$

$$D_h^{(lin)} = \left[d_{il}^{(lin)} = \int_\Omega \bar{S}\phi_i \nabla \cdot \Phi_l \, dx\right],$$

$$G_h^{(lin)} = \left[g_{li}^{(lin)} = \int_\Omega \nabla(\bar{S}\phi_i) \cdot \Phi_l \, dx\right].$$

- *Nonlinear/explicit*

$$A_h^{n,(nl)} = \left[a_{ij}^{n,(nl)} = \int_\Omega \tilde{\kappa}^n \nabla \phi_i \cdot \nabla \phi_j \, dx \right],$$

$$M_h^{n,(nl)} = \left[m_{ij}^{n,(nl)} = \int_\Omega \tilde{c}^n \phi_i \phi_j \, dx \right],$$

$$K_h^{n,(nl)} = \left[b_{ln}^{n,(nl)} = \int_\Omega [2\tilde{\mu}^n \varepsilon(\Phi_l) : \varepsilon(\Phi_n) \, dx + \tilde{\lambda}^n \nabla \cdot \Phi_l \, \nabla \cdot \Phi_n] \, dx \right],$$

$$D_h^{n,(nl)} = \left[d_{il}^{n,(nl)} = \int_\Omega \tilde{S}^n \phi_i \nabla \cdot \Phi_l \, dx \right],$$

$$G_h^{n,(nl)} = \left[g_{li}^{n,(nl)} = \int_\Omega \nabla(\tilde{S}^n \phi_i) \cdot \Phi_l \, dx \right].$$

The implementation of the scheme (4.2) at each time iteration can be written as follows:

$$(4.5) \qquad \begin{bmatrix} M_h^{(lin)} + \tau A_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix} \begin{bmatrix} p_h^{n+1} \\ u_h^{n+1} \end{bmatrix} = \begin{bmatrix} b_{p,h}^n \\ b_{u,h}^n \end{bmatrix},$$

with

$$\begin{bmatrix} b_{p,h}^n \\ b_{u,h}^n \end{bmatrix} = \begin{bmatrix} \tau F_{p,h} \\ F_{u,h}^n - F_{u,h}^{n-1} \end{bmatrix} + \begin{bmatrix} M_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix} \begin{bmatrix} p_h^n \\ u_h^n \end{bmatrix}$$

$$- \begin{bmatrix} M_h^{n,(nl)} & \alpha D_h^{n,(nl)} \\ \alpha G_h^{n,(nl)} & K_h^{n,(nl)} \end{bmatrix} \begin{bmatrix} p_h^n - p_h^{n-1} \\ u_h^n - u_h^{n-1} \end{bmatrix} - \begin{bmatrix} \tau A_h^{n,(nl)} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_h^n \\ u_h^n \end{bmatrix}.$$

Here, at each iteration, we have a linear system, where the matrix does not depend on the current solution. Therefore, the matrix is constructed only once, and by pre-inverting/pre-factorizing or pre-initializing a solver, we can significantly accelerate the simulations.

5. Two-grid solver. This section outlines the construction of the two-grid solver for the unsaturated poroelasticity problem, focusing on its two main components: (1) the construction of the coarse-grid approximation, and (2) the design of the coupled smoothing iterations.

Using the proposed implicit–explicit scheme (4.5), we consider the following linear system on each time iteration for $y_h^{n+1} = (p_h^{n+1}, u_h^{n+1})$:

$$L_h^{(lin)} y_h^{n+1} = b_h^n,$$

where $L_h^{(lin)}$ is defined using the linear part of the operators,

$$L_h^{(lin)} = \begin{bmatrix} M_h^{(lin)} + \tau A_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix}, \quad \text{and} \quad b_h^n = \begin{bmatrix} b_{p,h}^n \\ b_{u,h}^n \end{bmatrix}.$$

In this work, we consider the following two-grid algorithm for a given initial guess $y^{(0)}$.

1. Coarse-grid correction:
   (a) Restriction: $r_H = P^T (b_h^n - L_h^{(lin)} y^{(0)})$.
   (b) Coarse-grid solution: $L_H^{(lin)} e_H = r_H$ with $L_H^{(lin)} = P^T L_h^{(lin)} P$
   (c) Interpolation and update: $y^{(1)} = y^{(0)} + P e_H$.
2. Post-smoothing: $y^{(2)} = y^{(1)} + S_h^{-1} (b_h^n - L_h^{(lin)} y^{(1)})$ and $y_h^{n+1} = y^{(2)}$.

Here, $S_h \in \mathbb{R}^{N_h \times N_h}$ is a smoothing operator and $L_H^{(lin)} \in \mathbb{R}^{N_H \times N_H}$ is the Galerkin coarse-grid matrix constructed using the prolongation operator $P \in \mathbb{R}^{N_h \times N_H}$ and the restriction operator $R = P^T \in \mathbb{R}^{N_H \times N_h}$. We note that, in this setting, we only use post-smoothing iterations [13, 33].

5.1. Coarse-grid solver. To construct a coarse-grid solver, we follow a similar approach as considered in the geometric multigrid method and apply a continuous Galerkin approximation on the coarse grid. We define the local domain ω_l as a combination of several coarse-grid cells K_i that share the same coarse-grid node ($l = 1, \ldots, N_v^H$, where N_v^H is the number of coarse-grid vertices). We denote the coarse grid with $\mathcal{T}^H = \bigcup_{i=1}^{N_c^H} K_i$ (see Figure 5.1). We note that the multiscale space is constructed in a preprocessing stage for a given heterogeneity setting [3, 8].
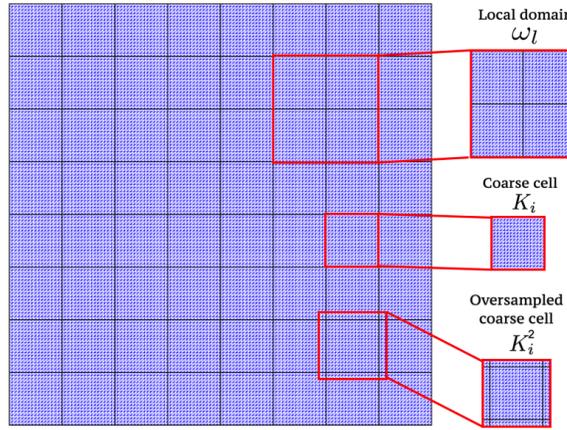

FIG. 5.1. *Fine grid (blue color), coarse grid (black color), local domain ω_l, coarse cell K_i and oversampled coarse cell K_i^o for $o = 2$ with local fine-grid resolution.*

Multiscale basis functions for pressure. We solve the following local spectral problem for $(\lambda_{p,j}, \phi_j^{\omega_l})$ in the local domain ω_l with a fine-scale resolution

$$(5.1) \qquad A_h^{(lin),\omega_l} \phi_j^{\omega_l} = \lambda_{p,j} S_h^{A,\omega_l} \phi_j^{\omega_l},$$

where

$$A_h^{(lin),\omega_l} = [a_{ij}], \quad a_{ij} = \int_{\omega_l} \bar{\kappa}(x) \nabla \phi_i \cdot \nabla \phi_j \, dx,$$

$$S_h^{A,\omega_l} = [s_{ij}^A], \quad s_{ij}^A = \int_{\omega_l} \bar{\kappa}(x) \phi_i \phi_j \, dx.$$

We choose eigenvectors $\phi_j^{\omega_l}$ ($j = 1, \ldots, M^{l,p}$) corresponding to the smallest $M^{l,p}$ eigenvalues in ω_l and multiply by linear partition-of-unity functions χ^l to obtain conforming basis functions

$$Q_{ms} = \text{span}\{\psi_j^{\omega_l}, \, l = 1, \ldots, N_v^H, \, j = 1, \ldots, M^{l,p}\},$$

where $\psi_j^{\omega_l} = \chi^l \phi_j^{\omega_l}$ and χ^l is the linear partition-of-unity function in ω_l.

Multiscale basis functions for displacements. We solve the following local spectral problem for $(\lambda_{u,j}, \Phi_j^{\omega_l})$ in the local domain with a fine-scale resolution

$$(5.2) \qquad K_h^{(lin),\omega_l} \Phi_j^{\omega_l} = \lambda_{u,j} S_h^{K,\omega_l} \Phi_j^{\omega_l},$$

where

$$K_h^{(lin),\omega_l} = [b_{ij}], \quad b_{ij} = \int_{\omega_l} \bar{\sigma}(\Phi_i) : \varepsilon(\Phi_j)\, dx,$$

$$S_h^{K,\omega_l} = [s_{ij}^K], \quad s_{ij}^K = \int_{\omega_l} (\bar{\lambda} + 2\bar{\mu}) \Phi_i \Phi_j\, dx.$$

We choose eigenvectors $\Phi_j^{\omega_l}$ ($j = 1, \ldots, M^{l,u}$) corresponding to the smallest $M^{l,u}$ eigenvalues and multiply by linear partition-of-unity functions to obtain conforming basis functions

$$V_{ms} = \mathrm{span}\{\Psi_j^{\omega_l}, \, l = 1, \ldots, N_v^H, \, j = 1, \ldots, M^{l,u}\},$$

where $\Psi_j^{\omega_l} = \chi^l \Phi_j^{\omega_l}$.

Coarse-grid system. Using constructed multiscale basis functions for pressure and displacements, we define the prolongation matrix as follows:

$$P = \begin{bmatrix} P_p & 0 \\ 0 & P_u \end{bmatrix}$$

where

$$P_p = (\psi_1^{\omega_1} \cdots \psi_{M^1,p}^{\omega_1} \cdots \psi_1^{\omega_{N_v^H}} \cdots \psi_{M^{N_v^H},p}^{\omega_{N_v^H}}),$$

$$P_u = (\Psi_1^{\omega_1} \cdots \Psi_{M^1,u}^{\omega_1} \cdots \Psi_1^{\omega_{N_v^H}} \cdots \Psi_{M^{N_v^H},u}^{\omega_{N_v^H}}).$$

We note that, for discretization in the local domain ω_l, we use the same approximation method as used in the global problem (3.1) and construct a map from local degrees of freedoms (DOFs) to global ones as the prolongation operator.

Therefore, we have

$$L_H^{(lin)} = P^T L_h^{(lin)} P = \begin{bmatrix} P_p^T & 0 \\ 0 & P_u^T \end{bmatrix} \begin{bmatrix} M_h^{(lin)} + \tau A_h^{(lin)} & \alpha D_h^{(lin)} \\ \alpha G_h^{(lin)} & K_h^{(lin)} \end{bmatrix} \begin{bmatrix} P_p & 0 \\ 0 & P_u \end{bmatrix}$$

$$= \begin{bmatrix} P_p^T(M_h^{(lin)} + \tau A_h^{(lin)})P_p & \alpha P_p^T D_h^{(lin)} P_u \\ \alpha P_u^T G_h^{(lin)} P_p & P_u^T K_h^{(lin)} P_u \end{bmatrix}$$

$$= \begin{bmatrix} M_H^{(lin)} + \tau A_H^{(lin)} & \alpha D_H^{(lin)} \\ \alpha G_H^{(lin)} & K_H^{(lin)} \end{bmatrix},$$

with

$$M_H^{(lin)} = P_p^T M_h^{(lin)} P_p, \quad A_H^{(lin)} = P_p^T A_h^{(lin)} P_p, \quad K_H^{(lin)} = P_u^T K_h^{(lin)} P_u,$$

$$D_H^{(lin)} = P_p^T D_h^{(lin)} P_u, \quad G_H^{(lin)} = P_u^T G_h^{(lin)} P_p.$$

We use a direct solver to solve the coarse-scale problem, assuming aggressive coarsening so that the resulting coarse system is small and can be inverted efficiently. The total number of coarse-scale unknowns is $N_H = N_H^p + N_H^u$, where $N_H^p = \sum_{l=1}^{N_v^H}(1 + M^{l,p})$ and $N_H^u = \sum_{l=1}^{N_v^H}(1 + M^{l,u})$. For smoothing iterations, we consider the classical pointwise smoothers, Jacobi and Gauss–Seidel, and further propose a coupled blockwise smoother.

5.2. Smoothing iterations. We start with defining subdomains based on the coarse grid that effectively balances computational efficiency with inter-subdomain communication, and reduces errors in a coupled way for our poroelasticity problem. In our approach, a coarse grid is aligned with coarse facets, such that $K_i \cap K_j = E_{ij}$. Then the computational domain Ω is divided into N_c^H subdomains $\{K_i\}_{i=1}^{N_c^H}$ with minimal overlap consisting of nodes located on the interfaces E_{ij}. We then increase overlap, considering the oversampled local domain K_i^o, which is constructed based on the coarse cell K_i by adding o layers of fine-mesh width h ($o = 1, 2$). The combination of overlapping Schwarz smoothers of this form with local spectral coarse grids is highly related to domain decomposition methods based on generalized eigenvalue problems in the overlap (GenEO) [31]. Last, we consider the case with the largest overlap defined by the support of coarse-grid basis functions ω_l. In Figure 5.1, we depicted local domains ω_l and K_i^o for $o = 0$ and $o = 2$. In the context of multigrid smoothing, we update the approximation in each iteration based on the local residual over a subdomain corresponding to either the coarse-grid cell $\iota_i = K_i$ or the support of the multiscale basis function $\iota_i = \omega_i$.

Let ι_i be the subdomain of global domain $\Omega = \bigcup_{i=1}^{N_\iota^H} \iota_i$, where $\iota_i = K_i$, K_i^o, or ω_i, with $N_H^\iota = N_c^H$ for $\iota_i = K_i, K_i^o$, and $N_H^\iota = N_v^H$ for $\iota_i = \omega_i$ (with N_c^H and N_v^H the number of coarse-grid cells and vertices, respectively). Then, we formulate the smoothing iterations as a Schwarz-type method, where we solve the following local coupled poroelasticity problems within each local domain ι_i:

$$L_h^{(lin),i} w_i^n = r_i^n.$$

Here $L_h^{(lin),i}$ is the restriction of the global system matrix $L_h^{(lin)}$ to the subdomain K_i, and r_i^n is the restriction of the global residual r^n to the local domain K_i, i.e.,

$$L_h^{(lin),i} = R_i L_h^{(lin)} R_i^T, \quad r_i^n = R_i r^n,$$

where R_i is the restriction operator that maps global degrees of freedom to local degrees of freedom in the ith subdomain. Due to overlaps at coarse cell interfaces or on bigger overlap, the global correction can be expressed as a weighted average at interface points,

$$w^n = \sum_{i=1}^{N_\iota^H} R_i^T W_i w_i^n,$$

where W_i is the weighting matrix to account for the overlapping degrees of freedom, with each diagonal entry equal to the reciprocal of the number of patches that contain the corresponding degree of freedom [15, 29].

Then, the smoothing iterations can be represented as follows:

$$y^{(2)} = y^{(1)} + S_h^{-1} r^n = y^{(1)} + w^n,$$

with $w^n = S_h^{-1} r^n$ and $r^n = b_h^n - L_h^{(lin)} y^{(1)}$, where $y^{(1)}$ is the given approximation on the fine-grid solver and the global smoothing operator S_h^{-1} is defined as

$$S_h^{-1} = \sum_{i=1}^{N_\iota^H} R_i^T W_i (L_h^{(lin),i})^{-1} R_i.$$

The smoothing iterations can therefore be formulated as

$$(5.3) \qquad y^{(2)} = y^{(1)} + \sum_{i=1}^{N_\iota^H} R_i^T W_i (L_h^{(lin),i})^{-1} R_i (b_h^n - L_h^{(lin)} y^{(1)}),$$

which corresponds to the additive type of smoothing iterations in subdomains.

We focus on the structured uniform coarse grid and define four groups of subdomains, $J^k = \bigcup_{i \in I_k} \iota_i$ $(k = 1, \ldots, 4)$, based on indexing groups I_k for a colored patch relaxation. This adds a level of sequential solving to the relaxation, as patches in group $k + 1$ will relax with an updated residual following relaxation on patches in group k. In Figure 5.2, we plot a coarse grid (black lines) and groups of subdomains for $\iota_i = \omega_i$ (maximal overlap) and $\iota_i = K_i^o$ (minimal overlap). We use four different colors to represent these groups in Figure 5.2, with each colored point indicating the center of a subdomain (either a coarse node or a coarse cell) belonging to a specific index set I_k.
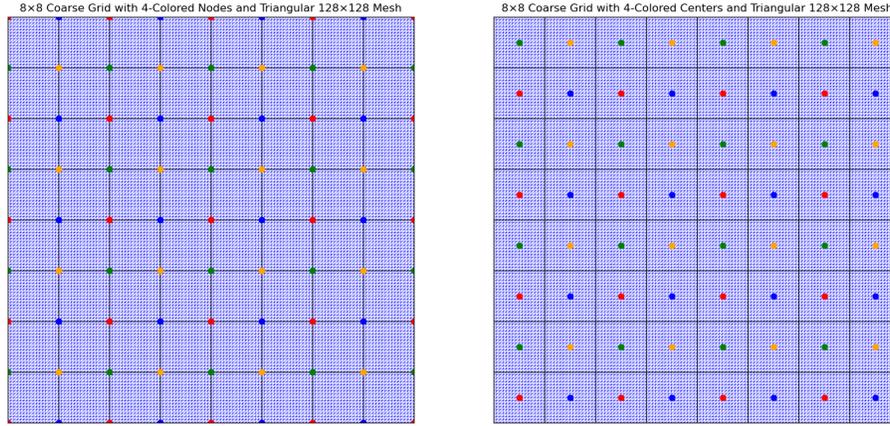

FIG. 5.2. *Coarse grid and groups of subdomains J^k $(k = 1, \ldots, 4)$. Left panel: $\iota_i = \omega_i$. Right panel: $\iota_i = K_i^o$.*

In the additive approach (5.3), updates are performed in each subdomain independently in a parallel manner, making it highly scalable with N_ι^H independent local solves. We can also formulate the smoothing iterations in a multiplicative way, where updates are applied sequentially in each subgroup using the latest available solution from the previous groups, leading to faster convergence than for the additive process. Moreover, the sequential approach is scalable in each group of subdomains, where calculations within the group can be done in parallel with $N_\iota^H/4$ independent local solves. The sequential iterations over groups can be represented as a fractional stepping [32],

$$y^{(1+k/M)} = y^{(1+(k-1)/M)} + \sum_{i \in I^k} R_i^T W_i (L_h^{(lin),i})^{-1} R_i (b_h^n - L_h^{(lin)} y^{(1+(k-1)/M)}),$$

where $k = 1, \ldots, M$ and $M = 4$ is the number of subdomain groups (colors).

6. Numerical results. We consider two numerical tests to evaluate the proposed splitting algorithm for the unsaturated poroelasticity problem in a heterogeneous domain.

Test 1 (splitting scheme and two-grid solver). We consider a fixed heterogeneous medium with prescribed contrast and model parameters in a two-dimensional setting. (*Test 1a*) First, we assess the accuracy of the proposed implicit–explicit splitting scheme by comparing it with an implicit time-stepping method employing nonlinear iterations. (*Test 1b*) Next, we investigate

the performance of the proposed multiscale two-grid solver with coupled smoothing iterations on three different fine-scale grids.

Test 2 (model parameters and computational cost). We study the performance of the proposed two-grid method under varying heterogeneity and model parameters. (*Test 2a*) First, we examine the effects of heterogeneity, contrast, nonlinearity, and boundary conditions on the performance of the two-grid method. (*Test 2b*) Next, we analyze in detail the offline (precomputation) and online computational costs of the proposed two-grid method for both two- and three-dimensional problems. In addition, we present results obtained without additional spectral basis functions, corresponding to the first constant eigenvector. In this case, the resulting coarse space consists of standard linear basis functions, making the method similar to the geometric multigrid-like approach. However, due to the aggressive coarsening strategy and the use of only two levels, this approach is not a classical geometric multigrid method.

For the two-grid solver, we consider the following:
- The implicit–explicit scheme with $N_t = 20$ time steps and fixed coarse-grid solver on an 8^d coarse grid, where d denotes the spatial dimension ($d = 2, 3$).
- The number of local spectral basis functions $M = 1, 2, 4, 8$, where local spectral basis functions for pressure and displacement are precomputed for a given heterogeneous field based on the linear setting described above.
- Smoothing iterations are chosen as follows: (i) Gauss–Seidel (GS) iterations from the pyamg library [6]; (ii) the proposed coupled smoothing iterations relative to the overlapping domain decomposition method, with local domains corresponding to the support of the multiscale basis functions ω_i (V); and (iii) coupled smoothing iterations with local domains corresponding to the coarse cells $K_i^0 = K_i$ (VK) and overlapped coarse cells including a few fine-cell layers K_i^o with $o = 1$ and 2 (VK1 and VK2).
- Simulations are performed without pre-smoothing iterations, and the number of post-smoothing iterations is varied from one to three.

The implementation is carried out in Python using the SciPy sparse library (version 1.16.2) to solve the linear systems at each time step [40]. Finite-element matrices are constructed using first-order continuous polynomial spaces for both the pressure and displacement variables. The finite-element implementation is based on the FEniCS library [23] (version 2019.1.0). To solve the eigenvalue problems arising in the construction of the multiscale basis functions, we use slepc4py, the Python interface to the SLEPc library [16] (version 3.23.3). For the iterative solver, we set the relative tolerance to 10^{-9} and the maximum number of iterations to 500. Simulations are run on a MacBook Pro with an Apple M2 Max chip and 32 GB RAM.

6.1. Test 1. We consider a two-dimensional formulation on the computational domain $\Omega = [0, L]^2$ with $L = 10$ m. The heterogeneous saturated hydraulic conductivity $k_s(x)$ [m^2] and the heterogeneous Young's modulus in the dry state $E_d(x)$ [Pa] are shown in Figure 6.1 (first and second panels), respectively.

In Figure 6.1, we plot a reference solution for an unsaturated poroelastic model with $N_t = 20$ on a 128×128 grid. We depict the pressure and the x and y components of the displacements at final time t_{20}. We can observe a strong influence of property heterogeneity on both pressure and displacements.

The model parameters are specified as follows:
- The van Genuchten model (2.2) and (2.3) is employed to describe the hydraulic nonlinearity, with $\nu = 0.37$, $\theta_r = 0.03$, $\theta_s = 0.45$, $\beta = 0.01$ [cm^{-1}], and $n_\theta = 1.6$, where $m_\theta = 1 - 1/n_\theta$. These parameters correspond to silt-type soils [24, 30]. For the nonlinear elastic model (2.4), we set $r_E = E_d/E_w = 2$ and $\zeta = 1.5$ [24].
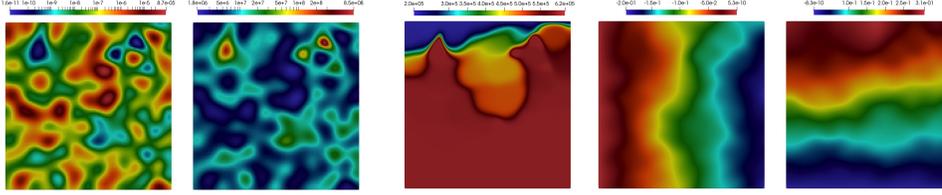
FIG. 6.1. *Test 1. Heterogeneous coefficients* $k_s(x)$ *[m^2] and* $E_d(x)$ *[Pa], pressure [Pa], and* x *and* y *components of displacement [cm] at the final time (from left to right).*

- The porosity is set to $\phi = 0.45$ and the Biot coefficient is set to $\alpha = 0.2$. The compressibilities of water and solid grains are taken as $C_w = C_s = 10^{-10}$ [Pa^{-1}]. We assume water density $\rho_w = 1000$ [kg m^{-3}] and gravitational acceleration $g = 9.8$ [m s^{-2}].
- For the flow problem, a Robin-type boundary condition is imposed on the top boundary with $p_1 = 202\,860$ [Pa] and $\gamma = 10^6$, while no-flux boundary conditions are applied on the remaining boundaries. For the mechanical problem, zero displacement is enforced in the x-direction on the left boundary and in the y-direction on the bottom boundary, whereas the remaining boundaries are assumed to be stress-free.
- The initial pore pressure is set to $p_0 = 602\,700$ [Pa]. Simulations are carried out over the time interval $t \in [0, T_{\max}]$ with $T_{\max} = 2$ days. The time-step size is defined as $\tau = T_{\max}/N_t$.

We consider the following numerical tests:

- *Test 1a (accuracy of the splitting scheme).* We assess the accuracy of the proposed implicit–explicit splitting scheme by comparing it with a fully implicit time-stepping method employing nonlinear iterations. The comparison is performed for varying numbers of time steps, $N_t = 10, 20, 40, 80$, on a fixed, uniformly refined 128×128 fine grid.
- *Test 1b (performance of the two-grid solver).* We investigate the performance of the proposed two-grid solver on $N \times N$ grids composed of triangular elements, with $N = 64, 128, 256$. Linear basis functions are used for both pressure and displacement variables. The corresponding total numbers of degrees of freedom are $\text{DOF}_h = 3(N+1)^2 = 12\,657, 49\,923$, and $198\,147$, respectively.

6.1.1. Test 1a (implicit–explicit scheme). First, we compare the performance of the three time integration schemes: (i) an implicit time-stepping scheme combined with Picard iterations to resolve the nonlinearity (3.2) (Im); (ii) a semi-implicit scheme, linearized about the previous solution (3.3) (sIm); and (iii) our proposed implicit–explicit scheme for linearization as in (4.2) (ImEx).

To compare two approaches for time integration, we calculate the relative error in L_2 norm,

$$e_p(p_{\mathrm{ref}}, p) = \frac{\|p_{\mathrm{ref}} - p\|}{\|p_{\mathrm{ref}}\|}, \quad e_u(u_{\mathrm{ref}}, u) = \frac{\|u_{\mathrm{ref}} - u\|}{\|u_{\mathrm{ref}}\|}, \quad \|p\|^2 = (p, p),$$

where p_{ref} and u_{ref} are the reference solutions using the implicit scheme and Picard iterations. In nonlinear iterations, we set $N_{nl} = 10$ as a maximum number of nonlinear iterations and iterate till $e_p(p_h^{n+1,(m+1)}, p_h^{n+1,(m)}) \leq 0.1\%$ and $e_u(u_h^{n+1,(m+1)}, u_h^{n+1,(m)}) \leq 0.1\%$.

In Figure 6.2, we plot the number of nonlinear iterations for an implicit scheme (Im) with Picard iterations in green color. We investigate the influence of the number of time steps

$N_t = 10, 20, 40$. For all tested N_t, more Picard iterations are required in the initial time steps, eventually stabilizing to two iterations per step. Larger N_t leads to a quicker reduction to two nonlinear iterations. We note that in the linearized scheme (sIm – blue, ImEx – red), only one iteration is performed per time step. In Figure 6.3, we represent errors at a final time for different N_t showing stable first-order convergence of both schemes. We observe almost the same errors for pressure and displacement for both linearization methods (sIm and ImEx), which decreased for a larger number of time steps as expected.
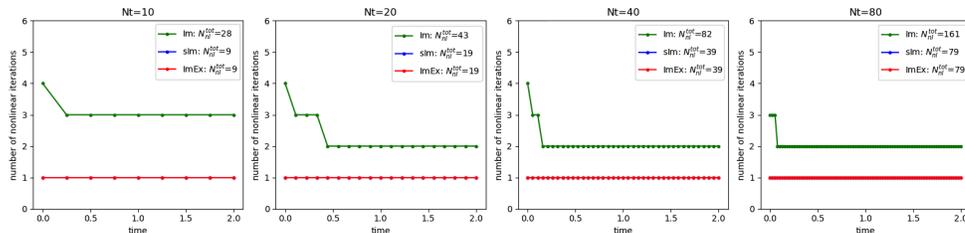

FIG. 6.2. *Number of nonlinear iterations for implicit scheme with Picard iterations. The legend shows the total number of nonlinear iterations* (N_{nl}^{tot}).
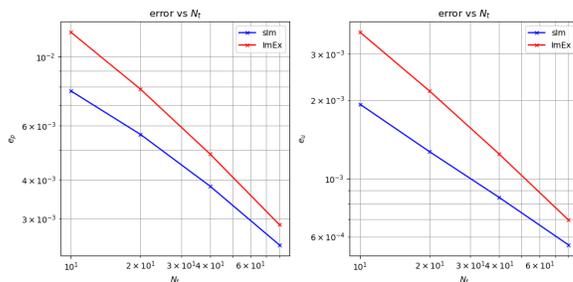

FIG. 6.3. *Number of time steps* (N_t) *versus relative error for pressure* e_p *(left) and displacements* e_u *(right) at final time.*

6.1.2. Test 1b (multiscale two-grid solver with coupled smoothing iterations). Next, we consider the performance of the proposed two-grid solver for an unsaturated poroelasticity problem in heterogeneous media. In the proposed approach with the additive splitting of the linear and nonlinear parts, we have the advantage of pre-initializing a solver for given heterogeneous properties. This includes the construction of a projection/prolongation operator, the construction of a coarse scale matrix based on the operator's linear part, and initializing a coupled smoother that includes pre-inverting local matrices in the smoothing algorithm based on the linear part of the operator.

We present results with three smoothing iterations for all types of smoothers, and the proposed coupled smoothing iterations are performed using a multiscale Vanka smoother with four colors and three post-smoothing iterations. In Figure 6.4, we investigate the convergence and the corresponding solve time. We show an average number of iterations of the linear solver and solve time (without "offline" time of generating system and initializing coarse solver) for a multiscale two-grid solver with different types of smoothers for the poroelasticity

problem on three grids ($N = 64, 128, 256$). We observe a clear dependence of the solution time on the number of basis functions, where increasing the number of basis functions from one to eight almost always leads to improved wall clock times. When we precompute the basis functions, the overhead cost of eight versus one is very small, so the best practice here is using more basis functions. All of the overlapping Vanka approaches reduce iteration counts by an order of magnitude or more on the finest grid compared with Gauss–Seidel, and demonstrate scalability in fine-mesh size, whereas Gauss–Seidel results in increased iteration count with mesh refinement. This is to some extent to be expected, because the size of the subdomains increases relative to fine-scale mesh with mesh refinement, but when the subdomain inverses can be precomputed and used across all times, such an approach does not raise significant scalability issues. In almost all cases, coarse cell-based Vanka with overlap of one or two cells is the best in terms of wall clock time. Using Vanka over the full subdomains $\{\omega_i\}$ decreases iterations versus Vanka based on coarse cells, but is working too hard in the sense that it increases wall clock time compared with cell-based Vanka.
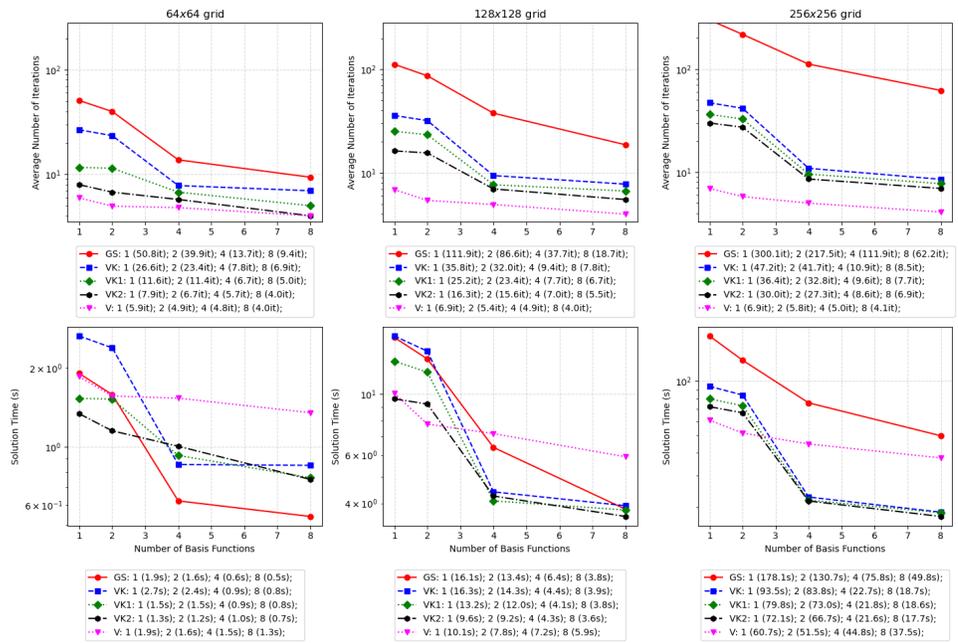

FIG. 6.4. *Average number of iterations of linear solver (first row) and solve time (second row) for a multiscale two-grid solver with different types of smoothers for 64×64, 128×128, and 256×256 grids (from left to right). GS, Gauss–Seidel; V, $H/2$ overlapped subdomains ω_i; VK, VK1 and VK2, minimal (interface), h, and 2h overlapped subdomains K_i^o with $o = 0, 1, 2$.*

In Figure 6.5, we present a visual comparative performance study of the VK2 smoother across different multicoloring strategies and coarse-grid resolutions. The first row represents the average number of linear iterations versus the number of multiscale basis functions for three grids (from left to right). The corresponding solve times (in seconds) are in the second row. Across all grids, the number of iterations decreases significantly as the number of multiscale basis functions M increases due to the better accuracy of the coarse space. We see that, for $M = 1$, all smoother configurations require more than 10 iterations. When we take a larger $M \geq 4$, most smoother configurations converge in about three or four iterations,

indicating that the solver becomes increasingly robust as the coarse space captures more global solution behavior. Among smoother variants, the four-color smoother configurations consistently converge with two to four iterations. Varying the number of smoothing iterations from one to three 3 improves convergence speed but increases solve time. We observe that multicoloring is essential in the proposed approach, where four-color smoothers consistently reduce both the number of iterations and wall clock time.
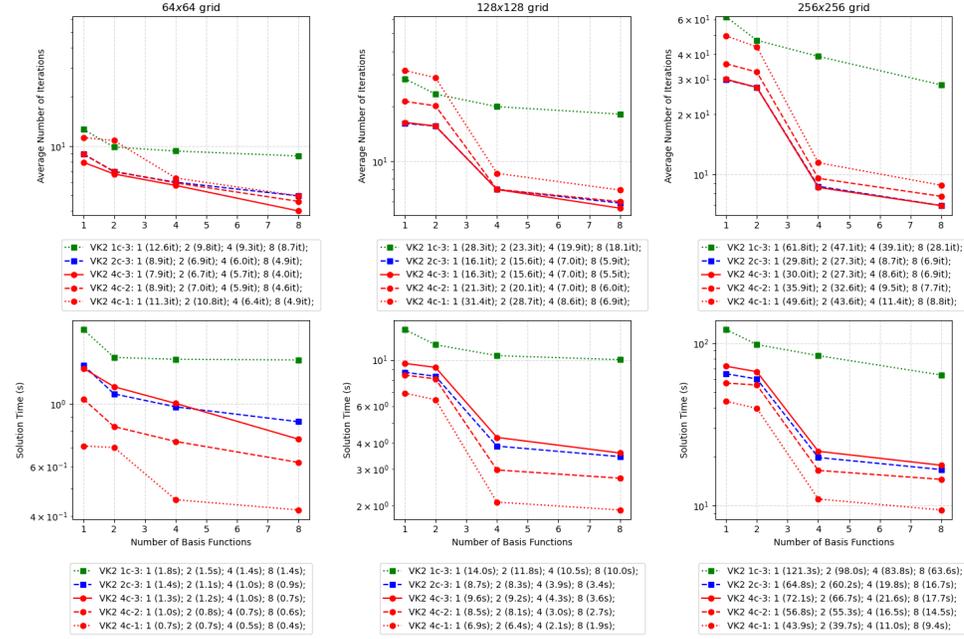

FIG. 6.5. *Performance of coupled smoother (VK2) with different multicoloring strategies and number of smoothing iterations (log scale). Results for three grids,* 64×64, 128×128, *and* 256×256 *(from left to right). Top: average iteration counts of linear solver. Bottom: corresponding solve times.*

6.2. Test 2. We consider both two- and three-dimensional formulations on the computational domains $\Omega = [0, L]^2$ and $[0, L]^3$ with $L = 1$ m. We investigate the performance of the proposed two-grid method under varying levels of heterogeneity, different model parameters, and different boundary conditions, through the following numerical tests:

- *Test 2a (model parameters).* We study the influence of heterogeneity, contrast, nonlinearity, and boundary conditions on the performance of the proposed two-grid method for a two-dimensional problem using a 128×128 fine grid ($N_h = 49\,923$) with $N_t = 20$ time steps (see Figures 6.6, 6.8, and 6.9).
- *Test 2b (computational cost).* We analyze both the offline (precomputation) and online computational costs of the proposed two-grid method for two- and three-dimensional problems (see Figures 6.6 and 6.7). In addition, we consider a coarse space constructed using standard linear basis functions only (i.e., without additional spectral basis functions). We consider 128×128 and 256×256 two-dimensional problems in *Test 2a* and the three-dimensional problem on a $32 \times 32 \times 32$ grid in *Test 2b*. The corresponding total numbers of degrees of freedom are $\mathrm{DOF}_h = 3(N + 1)^2 = 49\,923$, $198\,147$, and $143\,748$, respectively.
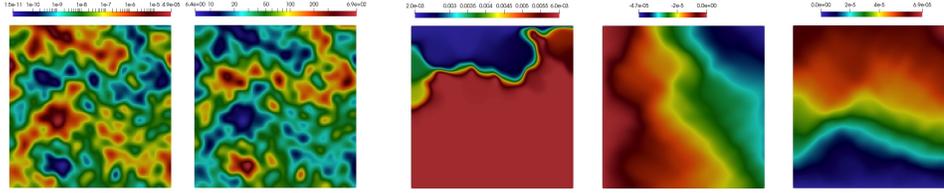
FIG. 6.6. *Test 2a. Heterogeneous coefficients $k_s(x)$ [m^2] and $E_d(x)$ [MPa], pressure [MPa], and x and y components of displacement [m] at final time (from left to right).*

FIG. 6.7. *Test 2b (3d). Heterogeneous coefficients $k_s(x)$ [m^2] and $E_d(x)$ [MPa], pressure [MPa], and x and y components of displacement [m] at final time (from left to right).*

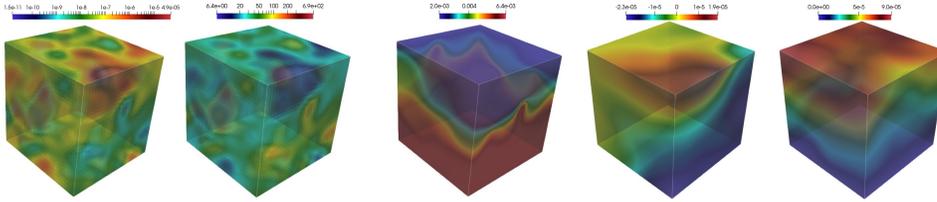The model parameters are specified as follows:
- The van Genuchten model (2.2) and (2.3) is used to describe the hydraulic nonlinearity, with parameters $\nu = 0.37$, $\theta_r = 0.03$, $\theta_s = 0.45$, $\beta = 1$ [m^{-1}], and $n_\theta = 1.6$, where $m_\theta = 1 - 1/n_\theta$. These values correspond to silt-type soils [24, 30]. For the nonlinear elastic model (2.4), we set $r_E = E_d/E_w = 2$ and $\zeta = 1.5$ [24].
- The porosity is set to $\phi = 0.45$, the Biot coefficient is set to $\alpha = 0.2$, and the compressibilities of water and solid grains are taken as $C_w = C_s = 10^{-3}$ [MPa^{-1}]. The water density is $\rho_w = 1000$ [kg m^{-3}], and the gravitational acceleration is $g = 9.8$ [m s^{-2}].
- For the flow problem, a Robin-type boundary condition is imposed on the top boundary with $p_1 = 0.0020286$ [MPa] and $\gamma = 100$, while no-flux conditions are applied on the remaining boundaries. For the mechanical problem, zero displacement is enforced in the x-direction on the left boundary and in the y-direction on the bottom boundary, whereas the remaining boundaries are assumed to be stress-free.
- The initial pressure is set to $p_0 = 0.006027$ [MPa]. Simulations are performed over the time interval $t \in [0, T_{\max}]$ with $T_{\max} = 2.31$ days.

In Figures 6.6 and 6.7, we depict the solutions for *Test 2a* and *Test 2b* using the parameters described above.

6.2.1. Test 2a (model parameters variation). In *Test 2a*, we examine the effects of heterogeneity, contrast, nonlinear parameters, and boundary conditions on the performance of the method using a fixed 128×128 grid. Taking *Test 2a* as a baseline, Figure 6.8 presents two additional tests.

In *Test 2a-bc*, we reduce the boundary coefficient to $\gamma = 0.1$ to investigate the influence of boundary conditions by varying the parameter associated with the Robin-type boundary condition. The default value $\gamma = 100$ in (2.7) corresponds to Dirichlet-type conditions, whereas the smaller value $\gamma = 0.1$ represents flux-related (Neumann-type) boundary conditions. In *Test 2a-nl*, we study the influence of nonlinear parameters on the solution and the performance of the two-grid method. The default setting corresponds to silt-type rocks, while in this additional test we consider a nonlinear model corresponding to sand. In the van Genuchten model (2.2)
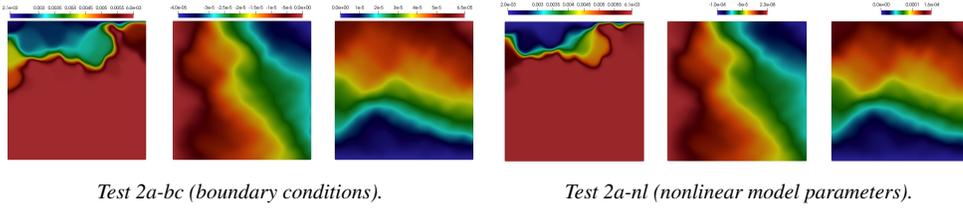
Test 2a-bc (boundary conditions).   *Test 2a-nl (nonlinear model parameters).*

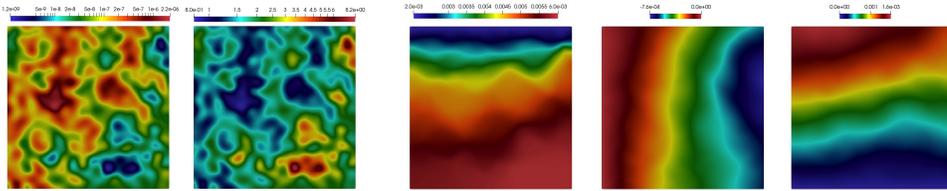FIG. 6.8. *Pressure [MPa], and u_x and u_y components of displacement [m] at the final time (from left to right).*

FIG. 6.9. *Test 2a-c (contrast). Heterogeneous coefficients $k_s(x)$ [m^2] and $E_d(x)$ [MPa], pressure [MPa], and x and y components of displacement [m] at final time (from left to right).*
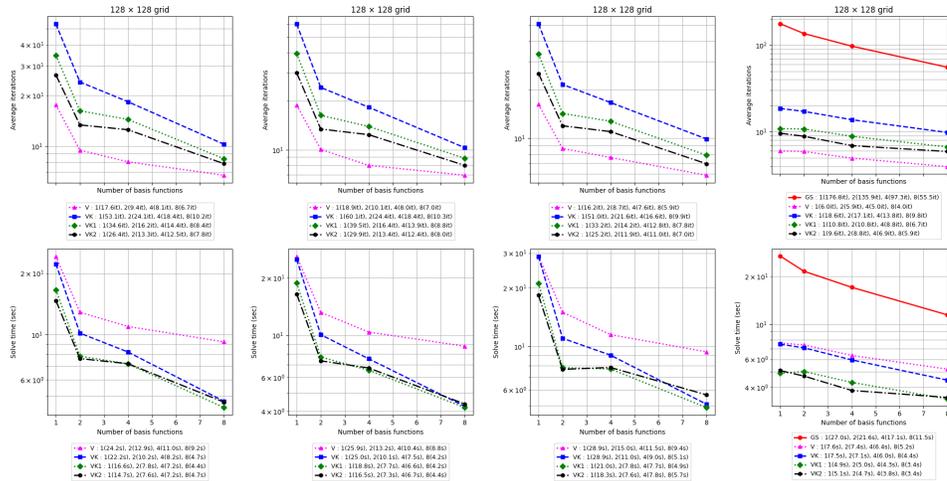
FIG. 6.10. *Tests 2a, 2a-bc, 2a-nl, and 2a-c (from left to right). Average number of iterations of linear solver (first row) and solve time (second row) for a multiscale two-grid solver with different types of smoothers for 128×128 grids (from left to right). GS, Gauss–Seidel; V, $H/2$ overlapped subdomains ω_i; VK, VK1, and VK2, minimal (interface), h, and 2h overlapped subdomains K_i^o with $o = 0, 1, 2$.*

and (2.3), the parameters are set as $\nu = 0.5$, $\theta_r = 0.018$, $\theta_s = 0.39$, $\beta = 0.035$ [m^{-1}], and $n_\theta = 2.52$, with $m_\theta = 1 - 1/n_\theta$ [24, 30]. For the nonlinear elastic model (2.4), we set $r_E = E_d/E_w = 2$ and $\zeta = 0.2$ [24]. Additionally, the parameters $\alpha = 0.8$ and $\phi = 0.39$ are used. Furthermore, we investigate the effect of contrast on the convergence of the method in *Test 2a-c* (see Figure 6.9).

Figure 6.10 summarizes the robustness and efficiency of the proposed multiscale two-grid solver under variations in boundary conditions, nonlinear material parameters, and coefficient contrast. Across all variants of *Test 2a*, the iteration counts remain uniformly

bounded, demonstrating that the two-grid method is robust with respect to changes in boundary conditions (*Test 2a-bc*) as well as increased nonlinearities associated with sand-type parameters (*Test 2a-nl*). In particular, the coupled overlapping Schwarz/Vanka-type smoothers with enlarged overlap (VK1 and VK2) consistently achieve the smallest number of iterations. Although these smoothers incur a higher per-iteration cost, the total solve time remains competitive and is often reduced due to the significantly lower iteration counts. In *Test 2a-c*, decreasing the coefficient contrast leads to a reduction in the number of iterations, and solvers with colored oversampled smoothers preserve fast convergence. Moreover, for this lower-contrast setting, convergence is also observed for the simple pointwise smoother (GS). Overall, the results confirm that the proposed two-grid framework exhibits robust performance with respect to both coefficient contrast and nonlinear effects, achieving a favorable balance between iteration counts and wall clock time when appropriate overlapping smoothers are employed.

6.2.2. Test 2b (computational cost). Finally, we compare the performance of the proposed two-grid method for both two- and three-dimensional problems (see Figures 6.6 and 6.7). In addition, we consider a coarse space constructed using standard linear basis functions. Specifically, we examine two-dimensional problems on 128×128 and 256×256 grids in *Test 2a*, as well as a three-dimensional problem on a $32 \times 32 \times 32$ grid in *Test 2b*.

FIG. 6.11. *Test 2a on a 128×128 grid, Test 2a on a 256×256 grid, and Test 2b on a $32 \times 32 \times 32$ grid (from left to right). Average number of iterations of linear solver (first row) and solve time (second row) for a multiscale two-grid solver with different types of smoothers and one post-smoothing iteration (from left to right). GS, Gauss–Seidel; V, $H/2$ overlapped subdomains ω_i; VK, VK1, and VK2, minimal (interface), h, and 2h overlapped subdomains K_i^o with $o = 0, 1, 2$; GS(3), Gauss–Seidel with three post-smoothing iterations.*

Figure 6.11 compares the performance of the proposed multiscale two-grid solver for both two- and three-dimensional problems. The results indicate that simple pointwise Gauss–Seidel

smoothing is insufficient for robust convergence in the two-dimensional case and is effective only for relatively small three-dimensional problems. In contrast, overlapping Schwarz and Vanka-type smoothers significantly enhance robustness across all considered test cases. In particular, the multiscale Vanka smoothers with enlarged overlap (VK1 and VK2) consistently achieve the smallest iteration counts across all tested configurations, leading to a reduction in overall solve time due to faster convergence. The advantages of increased overlap become more pronounced in three dimensions, where minimal or pointwise smoothers result in a substantial growth in iteration counts, whereas oversampled multiscale smoothers maintain stable and efficient convergence. These results demonstrate that an appropriate choice of coupled overlapping smoothers is essential for achieving scalable performance of the proposed multiscale two-grid framework.

Tables 6.1 and 6.2 report offline (precomputation) and online costs. The offline stage is dominated by solving the local generalized eigenvalue problems (5.1) and (5.2) for pressure and displacement. Table 6.1 (eight eigenvectors per subdomain) lists the average basis-construction time (tm_{loc}^{av}), average local problem size (DOF_{loc}^{av}), and the total offline time summed over subdomains (tm_{off}^{tot}). Because the local eigenproblems are independent, these computations are naturally parallelizable.
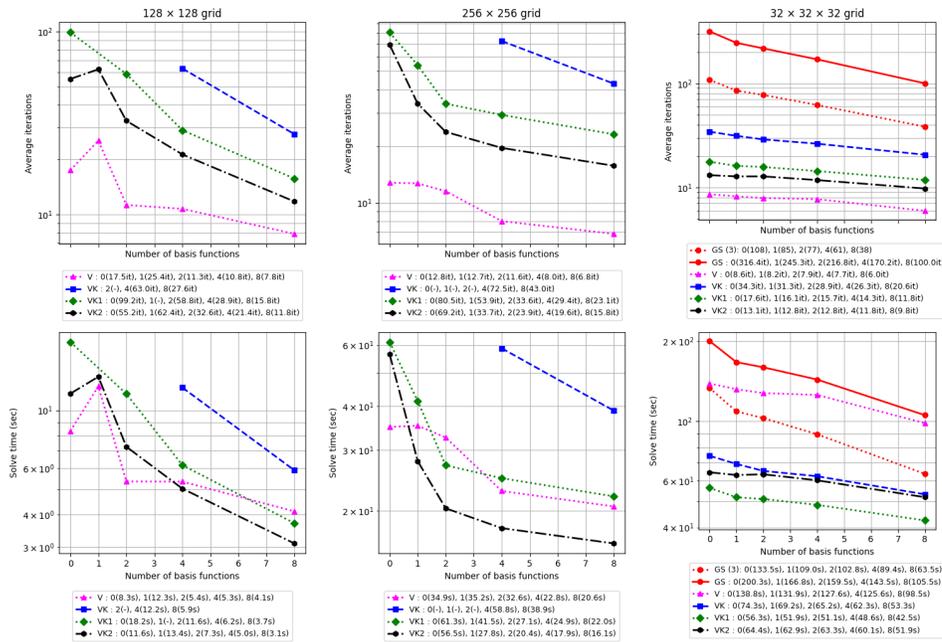
TABLE 6.1
Test 2a on a 128×128 grid, Test 2a on a 256×256 grid, and Test 2b on a $32 \times 32 \times 32$ grid (from left to right). Computational time of the offline stage (eight eigenvectors per local domain). Here tm_{loc}^{av}, DOF_{loc}^{av}, and tm_{off}^{tot} denote the average time (seconds) for solving the local eigenvalue problem (basis construction), the average size of the local problem, and the total time calculated as the sum over all local domains.

	128×128			256×256			$32 \times 32 \times 32$		
	tm_{loc}^{av}	DOF_{loc}^{av}	tm_{off}^{tot}	tm_{loc}^{av}	DOF_{loc}^{av}	tm_{off}^{tot}	tm_{loc}^{av}	DOF_{loc}^{av}	tm_{off}^{tot}
p	0.045	866.9	3.633	0.259	3351.12	21.024	0.507	2628.07	63.430
u	0.230	1733.9	18.600	1.936	6702.25	156.887	9.307	7884.22	1163.38

Table 6.2 compares online performance in two and three dimensions for standard linearized implicit (sIm), implicit–explicit (ImEx), and multiscale implicit–explicit (MsImEx) schemes with various solvers/smoothers. For sIm, Krylov solves (BiCGStab in scipy [40]) with smoothed-aggregation AMG preconditioning (pyamg [6]) require many iterations and long solve times, reflecting the difficulty of monolithic AMG solvers and resolving nonlinear coupling. ImEx reduces per-step time by fixing the matrix and no longer having to resolve the nonlinearity, but Krylov iteration counts remain similar. In contrast, MsImEx substantially lowers iteration counts and solve times by using an enriched multiscale coarse space with overlapping Schwarz and Vanka-type smoothers. Although MsImEx adds one-time offline setup (multiscale bases and local block solvers), these costs are amortized across nonlinear time steps since the precomputed components are reused. We also report the non-enriched coarse space ($M = 0$), as a loose proxy for geometric multigrid, which improves over sIm/ImEx but is less effective than spectral enrichment. Overall, once the offline stage is complete, MsImEx consistently converges in only a few iterations and delivers the lowest per-step solve times, especially for larger problems.

TABLE 6.2

Test 2a on a 128×128 grid, Test 2a on a 256×256 grid, and Test 2b on a $32 \times 32 \times 32$ grid (from left to right). Computational time and average size of local problem $(\overline{\overline{it}})$. Here tm_{in}, tm_{sys}, and tm_{sol} denote the cumulative wall clock times (seconds) for initialization, linear system generation, and linear system solution. The "—" symbol indicates that the method did not converge.

		128×128				256×256				$32 \times 32 \times 32$			
		tm_{in}	tm_{sys}	tm_{sol}	$\overline{\overline{it}}$	tm_{in}	tm_{sys}	tm_{sol}	$\overline{\overline{it}}$	tm_{in}	tm_{sys}	tm_{sol}	$\overline{\overline{it}}$
						sIm							
biCG+mg		—	18.38	54.07	213.0	—	69.15	399.87	419.0	—	2650.7	191.97	86.0
						ImEx							
biCG+mg		0.20	9.63	50.93	214.0	0.74	41.43	404.55	415.0	2.41	749.22	184.00	83.0
						MsImEx, $M = 0$							
GS	1	—	—	—	—	—	—	—	-	2.36	713.83	200.29	316.4
	2	—	—	—	—	—	—	—	—	2.39	722.16	149.37	160.2
	3	—	—	—	—	—	—	—	—	2.36	721.87	133.49	108.3
V	1	0.81	11.29	8.34	17.5	5.80	43.24	34.90	12.8	154.05	723.80	138.76	8.6
	2	0.81	11.37	14.41	15.6	5.55	43.36	63.85	11.8	154.64	722.26	215.59	6.9
	3	0.67	11.48	20.06	14.6	5.63	43.61	92.99	11.3	154.38	722.70	318.63	6.8
VK	1	—	—	—	—	—	—	—	—	6.20	722.53	74.30	34.3
	2	0.32	11.39	18.09	62.8	1.35	43.23	100.62	78.9	6.12	722.17	78.04	19.6
	3	0.31	11.29	21.65	53.0	1.30	43.69	119.75	63.7	6.18	723.48	85.83	14.8
VK1	1	0.36	11.46	18.23	99.2	1.26	43.23	61.27	80.5	11.97	723.15	56.31	17.6
	2	0.36	11.32	16.61	50.4	1.44	43.38	84.31	60.2	11.81	723.16	71.11	11.8
	3	0.35	11.49	21.60	45.6	1.44	43.23	109.85	54.0	11.87	722.89	95.78	10.8
VK2	1	0.40	11.42	11.58	55.2	1.63	43.02	56.51	69.2	25.37	723.06	64.42	13.1
	2	0.40	11.31	17.24	45.2	1.59	43.57	84.13	55.0	25.41	722.63	93.48	9.8
	3	0.41	11.51	22.09	39.8	1.56	43.09	111.49	49.6	25.25	723.34	124.75	8.9
						MsImEx, $M = 8$							
GS	1	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	6.33	719.73	72.72	53.4
	3	—	—	—	—	—	—	—	—	6.30	714.10	63.54	38.3
V	1	1.11	11.00	4.12	7.8	6.90	44.03	20.61	6.8	156.57	714.33	98.46	6.0
	2	1.08	10.98	6.54	6.8	6.74	43.45	32.65	5.9	155.94	714.33	180.54	5.7
	3	1.07	11.03	9.22	6.7	6.55	43.23	47.93	5.8	155.94	714.70	231.91	5.0
VK	1	0.58	10.91	5.91	27.6	2.21	43.13	38.89	43.0	10.17	714.12	53.26	20.6
	2	0.55	11.07	4.50	13.2	2.31	42.99	34.69	23.9	10.05	714.49	54.05	12.2
	3	0.57	10.98	4.71	10.2	2.23	43.36	33.32	16.9	10.04	715.05	60.99	9.8
VK1	1	0.61	10.91	3.71	15.8	2.38	42.63	22.04	23.1	15.96	714.49	42.51	11.8
	2	0.60	10.97	3.54	9.4	2.31	42.54	19.93	12.9	15.78	714.64	56.95	8.8
	3	0.61	11.01	4.39	8.4	2.38	43.22	21.45	9.8	15.77	714.29	73.16	7.9
VK2	1	0.65	11.01	3.10	11.8	2.44	42.60	16.14	15.8	29.09	713.79	51.95	9.8
	2	0.63	11.08	3.62	8.4	2.41	42.02	15.40	9.2	29.10	714.44	76.10	7.8
	3	0.66	11.04	4.67	7.8	2.46	43.00	18.47	7.8	29.02	714.36	98.36	6.8

7. Conclusion. In this work, we consider a coupled nonlinear system of equations that describe an unsaturated flow problem in heterogeneous poroelastic media. We derive a standard implicit scheme for the coupled system and then propose an implicit–explicit formulation, in which the stiff components of the problem are represented by a single linear operator that is treated implicitly. Careful construction of the additive splitting approach is presented for the unsaturated poroelasticity problem with the van Genuchten nonlinearity model and power law for the elastic modulus. We show that the proposed implicit–explicit scheme is unconditionally stable in the linear stability sense and enables an efficient implementation. Next, based on the proposed linearization, we develop a multiscale two-grid solver with coupled overlapping Schwarz smoothing iterations. Accurate coarse-grid approximation is constructed based on an enrichment of the linear coarse space with local spectral basis functions for both pressure and elasticity equations. A coupled smoothing iteration is proposed based on the overlapping Vanka approach and defining blocks relative to a coarse grid (oversampled coarse cell and local support of multiscale basis functions). Although the block smoothers present a significant computational cost, similar to the multiscale basis functions, by combining with the implicit–explicit formulation we are able to precompute these quantities once "offline" for a full nonlinear simulation. A numerical study is presented for a nonlinear coupled test problem of unsaturated flow in heterogeneous poroelastic media.

Appendix A. Details on the mathematical model.
We have a mass conservation equation in terms of (S, p) [18, 38, 39]:

$$
\begin{aligned}
\rho_w f &= \frac{\partial}{\partial t}(\rho_w \phi S) + \nabla \cdot (\rho_w \phi S q_w) \\
(A.1) \qquad &\approx \phi S \frac{\partial \rho_w}{\partial t} + \rho_w S \frac{\partial \phi}{\partial t} + \phi \rho_w \frac{\partial S}{\partial t} + \rho_w \nabla \cdot (\phi S q_{ws}) + \rho_w \phi S \nabla \cdot q_s \\
&= \rho_w \left(\phi S C_w \frac{\partial p}{\partial t} + \phi \frac{\partial S}{\partial t} + \nabla \cdot q_w \right) + \rho_w S \left(\frac{\partial \phi}{\partial t} + \phi \nabla \cdot q_s \right),
\end{aligned}
$$

where C_w is the water compressibility, $q_{ws} = q_w - q_s$ is the velocity of the water with respect to the solids, and $q_w = \phi S q_{ws}$ is the Darcy velocity of the water phase,

$$
q_w = -k(x) \frac{k_{rw}(S)}{\mu_w} \nabla(p - \rho_w g).
$$

Here k is the intrinsic permeability tensor, μ_w is the water dynamic viscosity, and k_{rw} is the relative permeability that accounts for the simultaneous flow of water and air. In hydrology, Darcy's law is often expressed in terms of the pressure, where $h_w = (p - p_a)/(\rho_w g)$ is the water pressure head (relative to the atmospheric pressure p_a) and $k_w = \rho_w g \, k/\mu_w$ is the hydraulic conductivity at saturated conditions. Under Richards' assumption ($p_a = 0$), we obtain equation (2.1).

For the conservation of the solid phase, we have [38]

$$
\begin{aligned}
0 &= \frac{\partial}{\partial t}((1 - \phi)\rho_s) + \nabla \cdot ((1 - \phi)\rho_s q_s) \\
&\approx (1 - \phi)\frac{\partial \rho_s}{\partial t} - \rho_s \frac{\partial \phi}{\partial t} + (1 - \phi)\rho_s \nabla \cdot q_s \\
&= -\rho_s \left(\frac{\partial \phi}{\partial t} + \phi \nabla \cdot q_s \right) + (1 - \phi)\left(\frac{\partial \rho_s}{\partial \sigma_T} \frac{\partial \sigma_T}{\partial t} + \frac{\partial \rho_s}{\partial p_s} \frac{\partial p_s}{\partial t} \right) + \rho_s \nabla \cdot q_s.
\end{aligned}
$$

Here, we have

$$\frac{\partial \rho_s}{\partial \sigma_T} = -\frac{\rho_s C_s}{1 - \phi}, \quad \frac{\partial \rho_s}{\partial p_s} = -\frac{\rho_s C_s \phi}{1 - \phi},$$

$$\nabla \cdot q_s = \frac{\partial \varepsilon_v}{\partial t}, \qquad \frac{\partial \sigma_T}{\partial t} = \frac{1}{C_m} \nabla \cdot q_s - \alpha p \frac{\partial S}{\partial t} - \alpha S \frac{\partial p}{\partial t},$$

with $\varepsilon_v = \nabla \cdot u$, $p_s = Sp$, and $\alpha = 1 - C_s/C_m$, where C_s and C_m are the compressibility of the solid grains and the porous medium, respectively. Therefore, we obtain

(A.2)
$$\begin{aligned}
\frac{\partial \phi}{\partial t} + \phi \nabla \cdot q_s &= -\rho_s \left(\frac{\partial \phi}{\partial t} + \phi \nabla \cdot q_s \right) + (1 - \phi) \left(\frac{\partial \rho_s}{\partial \sigma_T} \frac{\partial \sigma_T}{\partial t} + \frac{\partial \rho_s}{\partial p_s} \frac{\partial p_s}{\partial t} \right) + \rho_s \nabla \cdot q_s \\
&= -C_s \left(\frac{\partial \sigma_T}{\partial t} + \phi \frac{\partial p_s}{\partial t} \right) + \frac{\partial \varepsilon_v}{\partial t} \\
&= -C_s \left(\frac{1}{C_m} \frac{\partial \varepsilon_v}{\partial t} - \alpha p \frac{\partial S}{\partial t} - \alpha S \frac{\partial p}{\partial t} + \phi p \frac{\partial S}{\partial t} + \phi S \frac{\partial p}{\partial t} \right) + \frac{\partial \varepsilon_v}{\partial t} \\
&= \alpha \frac{\partial \varepsilon_v}{\partial t} + C_s \left((\alpha - \phi) p \frac{\partial S}{\partial t} + (\alpha - \phi) S \frac{\partial p}{\partial t} \right).
\end{aligned}$$

By combining equations (A.1) and (A.2), we obtain

$$\begin{aligned}
f &= \phi S C_w \frac{\partial p}{\partial t} + \phi \frac{\partial S}{\partial t} + \nabla \cdot q_w + S \left(\frac{\partial \phi}{\partial t} + \phi \nabla \cdot q_s \right) \\
&= \phi S C_w \frac{\partial p}{\partial t} + \phi \frac{\partial S}{\partial t} + \nabla \cdot q_w + \alpha S \frac{\partial \varepsilon_v}{\partial t} + S C_s \left((\alpha - \phi) p \frac{\partial S}{\partial t} + (\alpha - \phi) S \frac{\partial p}{\partial t} \right) \\
&= (\phi C_w + (\alpha - \phi) C_s S) S \frac{\partial p}{\partial t} + (\phi + (\alpha - \phi) C_s S p) \frac{\partial S}{\partial t} + \alpha S \frac{\partial \varepsilon_v}{\partial t} + \nabla \cdot q_w.
\end{aligned}$$

Then, we have

$$c(x, p) \frac{\partial p}{\partial t} + \alpha S \frac{\partial \varepsilon_v}{\partial t} - \nabla \cdot (\kappa(x, p) \nabla (p - \rho_w g)) = f,$$

with

$$c(x, p) = (\phi C_w + (\alpha - \phi) C_s S) S + (\phi + (\alpha - \phi) C_s S p) S', \quad \kappa(x, p) = k(x) \frac{k_{rw}(S)}{\mu_w},$$

where $S = S(p)$ and $S' = \partial S / \partial p$.

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, 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.

[2] J. H. ADLER, Y. HE, X. HU, S. MACLACHLAN, AND P. OHM, *Monolithic multigrid for a reduced-quadrature discretization of poroelasticity*, SIAM J. Sci. Comput., 45 (2023), pp. S54–S81.

[3] I. Y. AKKUTLU, Y. EFENDIEV, M. VASILYEVA, AND Y. WANG, *Multiscale model reduction for shale gas transport in poroelastic fractured media*, J. Comput. Phys., 353 (2018), pp. 356–376.

[4] M. ANITESCU, F. PAHLEVANI, AND W. J. LAYTON, *Implicit for local effects and explicit for nonlocal effects is unconditionally stable*, Electron. Trans. Numer. Anal., 18 (2004), pp. 174–187.
https://etna.ricam.oeaw.ac.at/vol.18.2004/pp174-187.dir/pp174-187.pdf

[5] U. M. ASCHER, S. J. RUUTH, AND B. T. WETTON, *Implicit–explicit methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 797–823.

[6] N. BELL, L. N. OLSON, J. SCHRODER, AND B. SOUTHWORTH, *PyAMG: algebraic multigrid solvers in Python*, J. Open Source Softw., 8 (2023), Paper No. 5495, 5 pages.

[7] R. H. BROOKS, *Hydraulic properties of porous media*, Tech. Report, Colorado State University, Fort Collins, 1965.

[8] D. L. BROWN AND M. VASILYEVA, *A generalized multiscale finite element method for poroelasticity problems I: Linear problems*, J. Comput. Appl. Math., 294 (2016), pp. 372–388.

[9] M. BUKAC, W. LAYTON, M. MORAITI, H. TRAN, AND C. TRENCHEA, *Analysis of partitioned methods for the Biot system*, Numer. Methods Partial Differential Equations, 31 (2015), pp. 1769–1813.

[10] T. BUVOLI AND B. S. SOUTHWORTH, *A new class of Runge-Kutta methods for nonlinearly partitioned systems*, SIAM J. Sci. Comput., 47 (2025), pp. A2609–A2635.

[11] J. CARMELIET, D. DEROME, M. DRESSLER, AND R. A. GUYER, *Nonlinear poro-elastic model for unsaturated porous solids*, J. Appl. Mech., 80 (2013), Paper No. 020909, 9 pages.

[12] M. A. CELIA, E. T. BOULOUTAS, AND R. L. ZARBA, *A general mass-conservative numerical solution for the unsaturated flow equation*, Water Resource Res., 26 (1990), pp. 1483–1496.

[13] E. T. CHUNG, Y. EFENDIEV, AND W. T. LEUNG, *Residual-driven online generalized multiscale finite element methods*, J. Comput. Phys., 302 (2015), pp. 176–190.

[14] P. E. FARRELL, M. G. KNEPLEY, L. MITCHELL, AND F. WECHSUNG, *PCPATCH: software for the topological construction of multigrid relaxation methods*, ACM Trans. Math. Software, 47 (2021), Paper No. 25, 22 pages.

[15] C. GREIF AND Y. HE, *A closed-form multigrid smoothing factor for an additive Vanka-type smoother applied to the Poisson equation*, Numer. Linear Algebra Appl., 30 (2023), Paper No. e2500, 19 pages.

[16] V. HERNANDEZ, J. E. ROMAN, AND V. VIDAL, *SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems*, ACM Trans. Math. Software, 31 (2005), pp. 351–362.

[17] M. KAWAI, A. IDA, H. MATSUBA, K. NAKAJIMA, AND M. BOLTEN, *Multiplicative Schwartz-type block multi-color Gauss-Seidel smoother for algebraic multigrid methods*, in Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, ACM, New York, 2020, pp. 217–226.

[18] J. KIM, *Sequential Methods for Coupled Geomechanics and Multiphase Flow*, Ph.D. Thesis, Stanford University, Stanford, 2010.

[19] J. KIM, H. A. TCHELEPI, AND R. JUANES, *Stability and convergence of sequential methods for coupled flow and geomechanics: fixed-stress and fixed-strain splits*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 1591–1606.

[20] A. E. KOLESOV, P. N. VABISHCHEVICH, AND M. V. VASILYEVA, *Splitting schemes for poroelasticity and thermoelasticity problems*, Comput. Math. Appl., 67 (2014), pp. 2185–2198.

[21] W. LAYTON, H. TRAN, AND C. TRENCHEA, *Analysis of long time stability and errors of two partitioned methods for uncoupling evolutionary groundwater-surface water flows*, SIAM J. Numer. Anal., 51 (2013), pp. 248–272.

[22] 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.

[23] A. LOGG, K.-A. MARDAL, AND G. WELLS, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, Springer, Berlin, 2012.

[24] N. LU AND M. KAYA, *Power law for elastic moduli of unsaturated soil*, J. Geotech. Geoenv. Engineer., 140 (2014), pp. 46–56.

[25] P. LUO, C. RODRIGO, F. J. GASPAR, AND C. W. OOSTERLEE, *Multigrid method for nonlinear poroelasticity equations*, Comput. Vis. Sci., 17 (2015), pp. 255–265.

[26] A. MIKELIĆ AND M. F. WHEELER, *Convergence of iterative coupling for coupled flow and geomechanics*, Comput. Geosci., 17 (2013), pp. 455–461.

[27] L. PETZOLD, *Differential/algebraic equations are not ODEs*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 367–384.

[28] C. RODRIGO, F. J. GASPAR, X. HU, AND L. T. ZIKATANOV, *Stability and monotonicity for some discretizations of the Biot's consolidation model*, Comput. Methods Appl. Mech. Engrg., 298 (2016), pp. 183–204.

[29] S. SABERI, G. MESCHKE, AND A. VOGEL, *A restricted additive Vanka smoother for geometric multigrid*, J. Comput. Phys., 459 (2022), Paper No. 111123, 16 pages.

[30] S. SILTECHO, C. HAMMECKER, V. SRIBOONLUE, C. CLERMONT-DAUPHIN, V. TRELO-GES, A. ANTONINO, AND R. ANGULO-JARAMILLO, *Use of field and laboratory methods for estimating unsaturated hydraulic properties under different land uses*, Hydrol. Earth Sys. Sci., 19 (2015), pp. 1193–1207.

[31] N. SPILLANE, V. DOLEAN, P. HAURET, F. NATAF, C. PECHSTEIN, AND R. SCHEICHL, *Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps*, Numer. Math., 126 (2014), pp. 741–770.

[32] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive*

Schwarz preconditioning, SIAM J. Sci. Comput., 29 (2007), pp. 2402–2425.

[33] A. TYRYLGIN, M. VASILYEVA, D. AMMOSOV, E. T. CHUNG, AND Y. EFENDIEV, *Online coupled generalized multiscale finite element method for the poroelasticity problem in fractured and heterogeneous media*, Fluids, 6 (2021), Paper No. 298, 21 pages.

[34] R. UZUOKA AND R. I. BORJA, *Dynamics of unsaturated poroelastic solids at finite strain*, Intern. J. Numer. Anal. Meth. Geomech., 36 (2012), pp. 1535–1573.

[35] P. N. VABISHCHEVICH, *Additive Operator-Difference Schemes: Splitting Schemes*, de Gruyter, Berlin, 2014.

[36] M. TH. VAN GENUCHTEN, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, Soil Sci. Soc. America J., 44 (1980), pp. 892–898.

[37] S. P. VANKA, *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.

[38] J. VARELA, *Implementation of an MPFA/MPSA-FV Solver for the Unsaturated Flow in Deformable Porous Media*, Master's Thesis, University Bergen, Bergen, 2018.

[39] J. VARELA, S. E. GASDA, E. KEILEGAVLEN, AND J. M. NORDBOTTEN, *A finite-volume-based module for unsaturated poroelasticity*, in Advanced Modeling with the MATLAB Reservoir Simulation Toolbox, K-A. Lie and O. Møyner, eds., Cambridge University Press, Cambridge, 2021, pp. 515–548.

[40] P. VIRTANEN, R. GOMMERS, T. E. OLIPHANT, M. HABERLAND, T. REDDY, D. COURNAPEAU, E. BUROVSKI, P. PETERSON, W. WECKESSER, J. BRIGHT, ET AL., *Scipy 1.0: fundamental algorithms for scientific computing in Python*, Nat. Meth., 17 (2020), pp. 261–272.

[41] Y. WANG, J. LIU, G. EGBERT, R. GUO, J. LI, G. YANG, S. TIAN, AND H. CHEN, *Multicolor-MG: an efficient parallel multigrid code based on multi-color block-wise Gauss Seidel smoothers for 3D natural source electromagnetic forward modeling*, Geophys., 90 (2025), pp. F43–F52.