Journal of Computational Physics 409 (2020) 109312

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Fully implicit hybrid two-level domain decomposition algorithms for two-phase flows in porous media on 3D unstructured grids

Li Luo[a,*], Lulu Liu[b], Xiao-Chuan Cai[c,d], David E. Keyes[a]

[a] Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia
[b] Department of Mathematics, Nanjing University of Science and Technology, Nanjing, 210094, China
[c] Department of Mathematics, University of Macau, Macau, China
[d] Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 20 August 2019
Received in revised form 23 December 2019
Accepted 5 February 2020
Available online 7 February 2020

Keywords:
Two-phase flow in porous media
Two-level domain decomposition
Discontinuous Galerkin
Unstructured grid
Inexact Newton
Parallel scalability

ABSTRACT

Simulation of subsurface flows in porous media is difficult due to the nonlinearity of the operators and the high heterogeneity of material coefficients. In this paper, we present a scalable fully implicit solver for incompressible two-phase flows based on overlapping domain decomposition methods. Specifically, an inexact Newton-Krylov algorithm with analytic Jacobian is used to solve the nonlinear systems arising from the discontinuous Galerkin discretization of the governing equations on 3D unstructured grids. The linear Jacobian system is preconditioned by additive Schwarz algorithms, which are naturally suitable for parallel computing. We propose a hybrid two-level version of the additive Schwarz preconditioner consisting of a nested coarse space to improve the robustness and scalability of the classical one-level version. On the coarse level, a smaller linear system arising from the same discretization of the problem on a coarse grid is solved by using GMRES with a one-level preconditioner until a relative tolerance is reached. Numerical experiments are presented to demonstrate the effectiveness and efficiency of the proposed solver for 3D heterogeneous medium problems. We also report the parallel scalability of the proposed algorithms on a supercomputer with up to 8,192 processor cores.

1. Introduction

Simulation of flows in subsurface porous media plays an important role in the performance of petroleum reservoir and the assessment of groundwater contamination. The extended Darcy's law accounting for the properties of both fluid and media is often used to address multi-phase flow, and the complexity of this model lies in the interaction of various modeling features, including gravity, capillary pressure effects, heterogeneity of the absolute permeability, and relative permeability functions. Extra difficulties are induced by complex geometry, faults, channels and deviated wells, etc. In general, useful results for a large-scale geological model with complex heterogeneity are obtained by using numerical methods on grids with millions or even billions of points, accurate discretization schemes, and efficient parallel solution algorithms [18].

* Corresponding author.
E-mail addresses: li.luo@kaust.edu.sa (L. Luo), lulu.liu@njust.edu.cn (L. Liu), cai@cs.colorado.edu (X.-C. Cai), david.keyes@kaust.edu.sa (D.E. Keyes).

https://doi.org/10.1016/j.jcp.2020.109312
0021-9991/© 2020 Elsevier Inc. All rights reserved.

In this paper, we focus on the flow of two incompressible and immiscible phases as a basic model that consists of two coupled nonlinear partial differential equations with heterogeneous coefficients. For general reservoir models, it is important to design spatial discretization schemes that approximate accurately the physical quantities. Cell-centered finite difference schemes are popular and efficient on regular domains supported by many reservoir simulators [45]. However, their stability may degrade on unstructured grids that are necessary for complex geometries [20]. Other widely used discretization methods for subsurface systems include finite volume methods [28,41,46] and mixed finite element methods [11,24,60,62]. In the past decade, discontinuous Galerkin (DG) finite element methods have been shown to be competitive with respect to other standard methods for transport problems, such as single phase flows [47], miscible displacement [48] and reactive transport [53,54]. The advantages of these methods include high order of convergence, local mass conservation at the element level, and the capability to handle unstructured grids, as well as the potential to increase the ratio of floating point operations to memory access operations, which is important for modern computer architectures. Application of DG methods to multiphase flows has become attractive in recent years with work varying from different time integration schemes and different degrees of coupling in the nonlinear solver, which are briefly reviewed below.

One popular numerical scheme used in practice is IMPES (implicit pressure and explicit saturation) [4,12,13,33] in which a pressure equation is first solved and then the saturation is updated by an explicit time-stepping scheme. Since the saturation often changes faster than the pressure, in general, several small saturation time steps are performed immediately after a pressure time step. Enhanced versions of IMPES were proposed in [1,35,36,54] to improve the accuracy and stability by using a semi-implicit scheme for the saturation equation or by introducing a number of iterations in a single pressure-saturation time step. Eslinger [23] presented a decoupled method based on the local discontinuous Galerkin scheme for handling compressible fluids on uniform grids. In [29,30,44], decoupled solution using a mixed finite element method for the pressure equation and an explicit or semi-implicit DG method for the saturation equation were studied. In [36], decoupled DG methods with interior penalties and upwinding schemes were applied to the original formulation while the pressure equation is obtained by summing the discretized conservation equations of two phases. It is believed that the most stable scheme for subsurface multi-phase flows is the fully implicit method in which all the coupled nonlinear equations are solved simultaneously [16,42,56,65]. The first attempt to combine the implicit Euler method with coupled DG schemes for 2D two-phase flows was demonstrated in [20,21]. The authors designed a non-symmetric interior penalty Galerkin (NIPG) method for the total fluid conservation formulation without using any upwinding or slope limiting techniques. More recently, Bastian [5] presented a fully-coupled approach using the symmetric interior penalty Galerkin (SIPG) scheme for the spatial discretization of incompressible two-phase flows. In this work, we extend the fully implicit NIPG method introduced in [20] to 3D unstructured grids. A penalty parameter is used to penalize both wetting phase pressure and capillary pressure but not the saturation directly, which is important when handling discontinuous saturation in a more complex porous media [20].

The extension of the 2D work of [20] to 3D is not straightforward because the fully implicit DG discretization of two-phase flow leads to a large system of nonlinear equations, which requires the solution of several linear systems per time step, costing a considerable amount of compute time. Therefore, it is challenging and crucial to design efficient nonlinear and linear solvers for the resulting algebraic systems. Moreover, for large simulation in 3D, the use of supercomputers and scalable parallel algorithms is indispensable. Early research on the parallel implementation of DG for various applications can be found in [2,8,43,7]. Efforts have been made in employing Newton method and its variants to solve the two-phase flow problems [16,42,39,60–64]. Good candidates for solving the linear Jacobian system are preconditioned Krylov subspace methods [50]. When a large number of processor cores are used, a robust and scalable preconditioner is needed to keep the compute time at an acceptable level while maintaining the number of Krylov iterations. The constrained pressure residual (CPR) preconditioner is widely used in the community of reservoir simulation [15,38]. This is based on subblocks of the Jacobian matrix and uses an approximate pressure solve such as algebraic multigrid (AMG) [25] to constrain the residual of the full system, and thus is known as a physics-splitting approach. However, CPR-AMG may not work well when some source terms corresponding to complex physics destroy the elliptic properties exploited by AMG. In addition, the decoupling process tends to cause a conflict between the convergence of AMG and the convergence of the outer CPR iteration [25].

Alternatively, the class of Schwarz-type preconditioners [52,55] based on overlapping domain decomposition is considered as a physics-coupled approach, where all the variables associated with a subdomain are solved together in a coupled way. A one-level restricted additive Schwarz (RAS) method [10] was used in [60–63] to build the preconditioner for the Jacobian system arise from an active-set reduced-space version of Newton method for the two-phase flow on uniform grids. The RAS preconditioner is naturally suitable to parallel computing since communication occurs only between neighboring subdomains during the restriction process. Yang et al. [62,63] systematically studied several performance-related parameters in the preconditioner to achieve optimal performance for the simulation. It was reported that the number of linear iterations increases when a larger number of processor cores is used, leading to the degeneration of the parallel scalability. The situation would become worse when a higher order DG discretization is used, since the resulting Jacobian system has more non-zero elements and is often more ill-conditioned.

To improve upon the one-level preconditioner when using a large number of processor cores, in this paper, we introduce a hybrid two-level RAS preconditioner by applying a coarse grid solve to enhance the robustness and the scalability of the global linear solver. The coarse grid correction not only resolves the low frequency components of the error, which can be represented on the coarse grid, but also provides inter-subdomain communication for the fine-level preconditioner, and thereby improves the overall performance of the preconditioner. As far as we know, this is the first work that uses the two-level overlapping domain decomposition method for preconditioning the Jacobian systems arising from the fully implicit DG discretization of two-phase flow problems.

The remainder of this paper is organized as follows. In Section 2, the system of incompressible two-phase flow in porous media is presented and followed by a fully implicit DG finite element discretization. In Section 3, we discuss in detail the inexact Newton-Krylov solver with the one-level and two-level RAS preconditioners. Results for several numerical experiments with parallel performance data are reported in Section 4. Some concluding remarks are given in Section 5.

2. Mathematical model

2.1. Governing equations

We consider the immiscible displacement process of incompressible two-phase flows in a three-dimensional porous medium. Let Ω be a bounded domain in $\mathbb{R}^3$. The flow of the wetting phase (i.e., water) and non-wetting phase (i.e., oil) in Ω is described by Darcy's law and the saturation equation for each phase. We denote by the subscript $\alpha = w$ and $\alpha = n$ the wetting and non-wetting phase, respectively. The Darcy velocity for each phase is determined by

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha g \nabla D), \quad \alpha = w, n, \tag{1}$$

and the saturation equation for each phase satisfying the mass conservation is given by

$$\phi \frac{\partial s_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, \quad \alpha = w, n, \tag{2}$$

where $\mathbf{u}_\alpha$, s_α, p_α, ρ_α, q_α are, respectively, the velocity, saturation, pressure, density, and source of phase α. ϕ is the porosity of the porous media and $\mathbf{K}$ is the absolute permeability tensor. For heterogeneous porous media, they can be discontinuous in space and can vary over several orders of magnitude. g is the gravitational acceleration constant, and D is the depth at position (x, y, z). The mobility function λ_α is a ratio of the relative permeability $k_{r\alpha}(s_w)$ and the viscosity μ_α,

$$\lambda_\alpha = \frac{k_{r\alpha}(s_w)}{\mu_\alpha}, \quad \alpha = w, n.$$

The saturations of the two phases are constrained by

$$s_w + s_n = 1. \tag{3}$$

The relation between the wetting and non-wetting phase pressures is described by the capillary pressure [13,29],

$$p_c(s_w) = p_n - p_w. \tag{4}$$

The total mobility is expressed as $\lambda_t = \lambda_w + \lambda_n$, and we also define the total source $q_t = q_w + q_n$. Then, by summing the saturation equations (2) for both phases and considering (1), (3) and (4), we obtain the potential equation

$$\nabla \cdot (-\lambda_t \mathbf{K} \nabla p_w - \lambda_n \mathbf{K} \nabla p_c + (\lambda_n \rho_n + \lambda_w \rho_w) \mathbf{K} g \nabla D) = q_t, \qquad \text{in} \quad \Omega. \tag{5}$$

Substituting (1) to (2), the wetting-phase saturation equation becomes

$$\phi \frac{\partial s_w}{\partial t} + \nabla \cdot (-\lambda_w \mathbf{K}(\nabla p_w - \rho_w g \nabla D)) = q_w, \qquad \text{in} \quad \Omega. \tag{6}$$

In this paper, we consider the total fluid conservation formulation coupling (5) and (6) for the two-phase flow problem. In [20], this formulation was shown to be more robust than the two-phase conservation formulation (2) with respect to the choice of penalty factor in our discretization scheme to be introduced later. Boundary conditions and initial condition are required to close the system. Let $\partial\Omega = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_0$, where Γ_{in} denotes the inlet boundary, Γ_{out} denotes the outlet boundary, and $\Gamma_{in} \cap \Gamma_{out} = \emptyset$. $\Gamma_0 = \partial\Omega \setminus \{\Gamma_{in} \cup \Gamma_{out}\}$ is the impermeable boundary. The boundary conditions are stated as:

$$\begin{aligned}
&\mathbf{u}_w \cdot \mathbf{n} = f_w^{in}, \ \mathbf{u}_n \cdot \mathbf{n} = f_n^{in}, && \text{on} \quad \Gamma_{in}, \\
&p_w = p_w^{out}, \ \lambda_n \mathbf{K} \nabla p_c \cdot \mathbf{n} = 0, && \text{on} \quad \Gamma_{out}, \\
&\mathbf{u}_w \cdot \mathbf{n} = 0, \ \mathbf{u}_n \cdot \mathbf{n} = 0, && \text{on} \quad \Gamma_0,
\end{aligned}$$

where $\mathbf{n}$ is the unit outward normal vector, f_w^{in} and f_n^{in} are given flow rates at the inlet. The initial condition is given by

$$s_w|_{t=0} = s_w^0, \qquad \text{in} \quad \Omega. \tag{7}$$

The equations are coupled nonlinearly through the relative permeability and the capillary pressure that are given by ([29,39]):

$$k_{rw}(s_w) = s_e^{\beta}, \quad k_{rn}(s_w) = (1-s_e)^{\beta}, \quad p_c(s_w) = -B_c \log(s_e), \tag{8}$$

where β, B_c are positive parameters and s_e is the normalized saturation defined as $s_e = (s_w - s_{rw})/(1 - s_{rw} - s_{rn})$. Here s_{rw} and s_{rn} are residual saturations for the wetting and non-wetting phases.

2.2. Fully implicit discontinuous Galerkin finite element discretization

In this paper, we solve the system of (5) and (6) simultaneously using a fully coupled approach. The discretization is based on a backward Euler scheme in time and a non-symmetric interior penalty Galerkin (NIPG) finite element method in space.

Let $\Omega_h = \{E\}$ be a quasi-uniform grid of Ω consisting of N_E elements. We denote by Γ_h the set of faces in Ω_h and by e the face shared by two elements E_a and E_b (or $e \in \partial\Omega_h$). We associate with e a unit normal vector $\mathbf{n}_e$ directed from E_a to E_b ($a > b$). Then, we define the jump and average of a function f on e as

$$[f] = (f|_{E_a})|_e - (f|_{E_b})|_e, \quad \{f\} = \frac{1}{2}\left((f|_{E_a})|_e + (f|_{E_b})|_e\right). \tag{9}$$

We also denote the harmonic mean of f at e as

$$H(f) = \frac{2\,(f|_{E_a})|_e\,(f|_{E_b})|_e}{(f|_{E_a})|_e + (f|_{E_b})|_e}. \tag{10}$$

If $e \in \partial\Omega_h$, the above jump and averages of f on e reduce to $[f] = \{f\} = H(f) = (f|_E)|_e$, and the normal vector $\mathbf{n}_e$ coincides with the outward normal $\mathbf{n}$.

Given an integer $m \geq 0$, the discontinuous finite element space is

$$\mathcal{D}_m = \left\{\psi \in L^2(\Omega);\ \psi|_E \in \mathcal{P}_m(E), \forall E \in \Omega_h\right\},$$

where $\mathcal{P}_m(E)$ is the space of polynomials of maximum degree m. Let us denote by p_w^n and s_w^n the approximation of p_w and s_w at the nth time step, respectively, and Δt the time step size. To simplify upcoming extended derivations, we ignore the effect of gravity ($g = 0$). Denote by $(\cdot,\cdot)_E$ the $L^2(E)$-inner product and by $\langle\cdot,\cdot\rangle_e$ the $L^2(e)$-inner product. Then, the fully implicit discontinuous Galerkin finite element discretization of the coupled equations (5) and (6) is described as follows: given $(p_w^n, s_w^n) \in \mathcal{D}_m \times \mathcal{D}_m$, find $(p_w^{n+1}, s_w^{n+1}) \in \mathcal{D}_m \times \mathcal{D}_m$, such that for $\forall\ \psi \in \mathcal{D}_m$,

$$\mathcal{F}_p\left(p_w^{n+1}, s_w^{n+1}\right) = 0, \tag{11}$$

$$\mathcal{F}_s\left(p_w^{n+1}, s_w^{n+1}\right) = 0, \tag{12}$$

where

$$\mathcal{F}_p\left(p_w^{n+1}, s_w^{n+1}\right) = B_p + F_p + \tilde{F}_p + \hat{F}_p, \tag{13}$$

$$\mathcal{F}_s\left(p_w^{n+1}, s_w^{n+1}\right) = B_s + F_s + \tilde{F}_s + \hat{F}_s. \tag{14}$$

In (13)-(14), B_p and B_s are bulk integrals obtained from integration by parts:

$$B_p = \sum_{E\in\Omega_h} \left(\lambda_t(s_w^{n+1})\mathbf{K}\nabla p_w^{n+1} + \lambda_n(s_w^{n+1})\mathbf{K}\nabla p_c(s_w^{n+1}), \nabla\psi\right)_E - \sum_{E\in\Omega_h} (q_t, \psi)_E, \tag{15}$$

$$B_s = \sum_{E\in\Omega_h} \left(\lambda_w(s_w^{n+1})\mathbf{K}\nabla p_w^{n+1}, \nabla\psi\right)_E + \sum_{E\in\Omega_h} \left(\frac{\phi}{\Delta t}(s_w^{n+1} - s_w^n) - q_w, \psi\right)_E. \tag{16}$$

F_p and F_s are jump terms correspond to face integrals obtained by using the regularity of the exact solution and the boundary conditions [20]:

$$\begin{aligned} F_p = &\sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \left\langle\left\{-\lambda_t(s_w^{n+1})\mathbf{K}\nabla p_w^{n+1}\cdot\mathbf{n}_e\right\}, [\psi]\right\rangle_e + \sum_{e\in\Gamma_{in}} \left\langle f_w^{in} + f_n^{in}, \psi\right\rangle_e \\ &+ \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_{out}\cup\Gamma_0)} \left\langle\left\{-\lambda_n(s_w^{n+1})\mathbf{K}\nabla p_c(s_w^{n+1})\cdot\mathbf{n}_e\right\}, [\psi]\right\rangle_e, \end{aligned} \tag{17}$$

$$F_s = \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \left\langle\left\{-\lambda_w(s_w^{n+1})\mathbf{K}\nabla p_w^{n+1}\cdot\mathbf{n}_e\right\}, [\psi]\right\rangle_e + \sum_{e\in\Gamma_{in}} \left\langle f_w^{in}, \psi\right\rangle_e. \tag{18}$$

$\tilde{F}_p$ and $\tilde{F}_s$ are additional terms for the purpose of stabilization; they vanish for the exact solution:

$$\tilde{F}_p = \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \left\langle \left\{\lambda_t(s_w^{n+1})\mathbf{K}\nabla\psi\cdot\mathbf{n}_e\right\}, \left[p_w^{n+1}\right]\right\rangle_e - \sum_{e\in\Gamma_{out}} \left\langle \lambda_t(s_w^{n+1})\mathbf{K}\nabla\psi\cdot\mathbf{n}_e, p_w^{out}\right\rangle_e$$

$$+ \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_{out}\cup\Gamma_0)} \left\langle \left\{\lambda_n(s_w^{n+1})\mathbf{K}\nabla\psi\cdot\mathbf{n}_e\right\}, \left[p_c(s_w^{n+1})\right]\right\rangle_e, \tag{19}$$

$$\tilde{F}_s = \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \left\langle \left\{\lambda_w(s_w^{n+1})\mathbf{K}\nabla\psi\cdot\mathbf{n}_e\right\}, \left[p_w^{n+1}\right]\right\rangle_e - \sum_{e\in\Gamma_{out}} \left\langle \lambda_w(s_w^{n+1})\mathbf{K}\nabla\psi\cdot\mathbf{n}_e, p_w^{out}\right\rangle_e. \tag{20}$$

Lastly, $\hat{F}_p$ and $\hat{F}_s$ are penalty terms used to constrain the weak continuity of the pressure:

$$\hat{F}_p = \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \gamma H\left(\|\lambda_t(s_w^{n+1})\mathbf{K}\|_\infty\right)\left\langle\left[p_w^{n+1}\right],[\psi]\right\rangle_e - \sum_{e\in\Gamma_{out}} \gamma H\left(\|\lambda_t(s_w^{n+1})\mathbf{K}\|_\infty\right)\left\langle p_w^{out},\psi\right\rangle_e$$

$$+ \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_{out}\cup\Gamma_0)} \gamma H\left(\|\lambda_t(s_w^{n+1})\mathbf{K}\|_\infty\right)\left\langle\left[p_c(s_w^{n+1})\right],[\psi]\right\rangle_e, \tag{21}$$

$$\hat{F}_s = \sum_{e\in\Gamma_h\setminus(\Gamma_{in}\cup\Gamma_0)} \gamma H\left(\|\lambda_w(s_w^{n+1})\mathbf{K}\|_\infty\right)\left\langle\left[p_w^{n+1}\right],[\psi]\right\rangle_e - \sum_{e\in\Gamma_{out}} \gamma H\left(\|\lambda_w(s_w^{n+1})\mathbf{K}\|_\infty\right)\left\langle p_w^{out},\psi\right\rangle_e. \tag{22}$$

The penalty factor γ is an important parameter for the performance of the method. We consider the definition in [5] that accounts for the space dimension d, polynomial degree m, and element size, as follows:

$$\gamma = \begin{cases} \sigma\dfrac{m(m+d-1)|e|}{\min(|E_a|,|E_b|)}, & \text{on} \quad \Gamma_h\setminus(\Gamma_{in}\cup\Gamma_{out}\cup\Gamma_0), \\ \sigma\dfrac{m(m+d-1)|e|}{|E|}, & \text{on} \quad \Gamma_h\cap\Gamma_{out}, \end{cases}$$

where σ is a user-defined parameter and some typical choices for various problems are available in [1,5,20,22]. For general problems with heterogeneous porous media, γ is effectively scaled by the magnitude of the permeability. In order to obtain the correct front propagation in the case of discontinuous initial conditions [22], we adopt $H\left(\|\lambda_t(s_w^{n+1})\mathbf{K}\|_\infty\right)$ and $H\left(\|\lambda_w(s_w^{n+1})\mathbf{K}\|_\infty\right)$ for scaling γ in (21) and (22), respectively.

Remark 2.1. The use of equal polynomial order for the approximations for the pressure and the saturation is natural and easy to implement [1,22]. We refer to [20] for the performance study using different polynomial orders for the two variables.

3. Inexact Newton-Krylov solver

The fully implicit temporal discretization results in a nonlinear algebraic system

$$\mathcal{F}(\mathbf{x}) = \mathbf{0} \tag{23}$$

to be solved at each time step, where $\mathbf{x}$ is the vector of unknowns. We use a 2×2 point-block ordering for $\mathbf{x}$ and $\mathcal{F}(\mathbf{x})$ in our implementation, in which the unknown values of pressure and saturation associated with each grid point are always together in the same small block. This improves the parallel efficiency in load balance as well as the cache performance [57]. We note that $\mathcal{F}$ is a highly nonlinear function, where the nonlinearities come from the relative permeability $k_{r\alpha}(s_w)$ and the capillary pressure function $p_c(s_w)$. Extra difficulties for solving (23) are induced by the heterogeneity of the permeability and the porosity.

In order to solve (23) efficiently, we consider the family of Newton-Krylov algorithms [9,27,34]. In the algorithm, an inexact Newton method with backtracking (INB) is employed as the outer iteration, and a preconditioned GMRES method is applied as the Jacobian solve. More precisely, the solution of the previous time step is taken as the initial guess $\mathbf{x}_0 = \mathbf{x}^n$, then the next approximate solution $\mathbf{x}_{k+1}$ is obtained by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{s}_k, \quad k = 0, 1, \ldots \tag{24}$$

Here λ_k is the step length determined by a line search procedure [17]. The Newton correction $\mathbf{s}_k$ is approximated by solving a right-preconditioned Jacobian system with a restarted GMRES [50],

$$J_k M_k^{-1}(M_k\mathbf{s}_k) = -\mathcal{F}(\mathbf{x}_k). \tag{25}$$

The Jacobian matrix $J_k = \mathcal{F}'(\mathbf{x}_k)$ is calculated at the current approximate solution $\mathbf{x}_k$ and M_k is the preconditioner to be presented later. In INB, the linear solver for (25) is stopped if

$$\|J_k\mathbf{s}_k + \mathcal{F}(\mathbf{x}_k)\| \le \eta_k\|\mathcal{F}(\mathbf{x}_k)\|, \tag{26}$$

which means that the linear system does not need to be solved exactly. Compared to the classical exact Newton method, the inexact treatment of the linear system can reduce substantially the computational cost [27]. To enhance the robustness of INB, we pick the forcing term η_k based on norms that are by-products of the iteration, as suggested by Eisenstat and Walker [19]: start with any $\eta_0 \in [0, 1)$, for $k = 1, 2, \dots,$ we choose

$$\eta_k = \frac{\left| \|\mathcal{F}(\mathbf{x}_k)\| - \| J_{k-1}\mathbf{s}_{k-1} + \mathcal{F}(\mathbf{x}_{k-1})\| \right|}{\|\mathcal{F}(\mathbf{x}_{k-1})\|}. \tag{27}$$

3.1. *Construction of the Jacobian*

The Jacobian matrix J_k is a key component in INB. In this study, we choose to analytically calculate J_k using the chain rule since the exact Jacobian matrix brings added robustness. For programming in the finite element framework, it is understood that the global Jacobian matrix J_k is constructed by adding successively the contribution of each element Jacobian matrix. In order to explicitly define the element Jacobian J^E, we first let $\left\{\psi_i^E : 1 \le i \le n_\psi\right\}$ be the basis for the discrete space $\mathcal{D}_m$, where n_ψ is the number of basis functions in element E. Note that the functions ψ_i^E are identically zero outside the element E. Thus, we can write

$$p_w^{n+1}|_E = \sum_{i=1}^{n_\psi} p_{w,i}^E \psi_i^E, \quad s_w^{n+1}|_E = \sum_{i=1}^{n_\psi} s_{w,i}^E \psi_i^E, \quad \forall\, E \in \Omega_h. \tag{28}$$

By inserting (28) to (13)-(14), we obtain the element residual vector in the general form:

$$\mathcal{F}^E\left(P_w^E, S_w^E\right) = \begin{pmatrix} \mathcal{F}_p^E \\ \mathcal{F}_s^E \end{pmatrix}, \tag{29}$$

where $P_w^E = \left(p_{w,i}^E\right)_{1\le i\le n_\psi}$ and $S_w^E = \left(s_{w,i}^E\right)_{1\le i\le n_\psi}$ are vectors of unknowns for p_w^{n+1} and s_w^{n+1} on element E. We denote by $\mathcal{F}_{p,i}^E$ (respectively $\mathcal{F}_{s,i}^E$) the row of $\mathcal{F}_p^E$ (respectively $\mathcal{F}_s^E$) corresponding to the test function ψ_i^E, with $1 \le i \le n_\psi$. Then, J^E can be written in a block form:

$$J^E = \begin{pmatrix} \frac{\partial \mathcal{F}_{p,i}^E}{\partial p_{w,j}^E} & \frac{\partial \mathcal{F}_{p,i}^E}{\partial s_{w,j}^E} \\ \frac{\partial \mathcal{F}_{s,i}^E}{\partial p_{w,j}^E} & \frac{\partial \mathcal{F}_{s,i}^E}{\partial s_{w,j}^E} \end{pmatrix}, \quad 1 \le i, j \le n_\psi. \tag{30}$$

When using the discontinuous Galerkin finite element discretization in space, the internal face integral in the element residual $\mathcal{F}^E$ is obtained from two neighboring elements. Therefore, the partial derivative of $\mathcal{F}_{p,i}^E$ (respectively $\mathcal{F}_{s,i}^E$) with respect to unknowns from the neighboring element should be taken into account. Assume that the internal face e is shared by two elements E_a and E_b. We take the term $\frac{\partial \mathcal{F}_{p,i}^E}{\partial s_{w,j}^E}$ as an example and show briefly how to calculate the internal face integrals for the Jacobian entries that are contributed from E_a and E_b, respectively. To simplify the notations, we define $\psi_i^l = \psi_i^{E_l}$, $p_w^l = p_w^k|_{E_l}$, $s_w^l = s_w^k|_{E_l}$, and $\mathbf{K}^l = \mathbf{K}|_{E_l}$, for $l = a, b$. The scaling factor for the penalty terms is approximated by $H\left(\|\lambda_t(s_w^k|_E)\mathbf{K}|_E\|_\infty\right)$. The analytical formulations for the corresponding derivatives are derived using the chain rule, as follows:

1. Internal face integral for the derivative of $\mathcal{F}_{p,i}^{E_a}$ with respect to unknowns of the saturation on the current element E_a:

$$\frac{\partial \mathcal{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} = \frac{\partial F_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} + \frac{\partial \tilde{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} + \frac{\partial \hat{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}}, \tag{31}$$

where

$$\begin{aligned} \frac{\partial F_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} &= -\frac{1}{2}\left\langle \lambda_t'(s_w^a)\psi_j^a \mathbf{K}^a \nabla p_w^a \cdot \mathbf{n}_e, \psi_i^a \right\rangle_e - \frac{1}{2}\left\langle \lambda_n'(s_w^a)\psi_j^a \mathbf{K}^a p_c'(s_w^a) \nabla s_w^a \cdot \mathbf{n}_e, \psi_i^a \right\rangle_e \\ &- \frac{1}{2}\left\langle \lambda_n(s_w^a)\mathbf{K}^a p_c''(s_w^a)\psi_j^a \nabla s_w^a \cdot \mathbf{n}_e, \psi_i^a \right\rangle_e - \frac{1}{2}\left\langle \lambda_n(s_w^a)\mathbf{K}^a p_c'(s_w^a) \nabla \psi_j^a \cdot \mathbf{n}_e, \psi_i^a \right\rangle_e, \end{aligned} \tag{32}$$

$$\begin{aligned} \frac{\partial \tilde{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} &= \frac{1}{2}\left\langle \lambda_n'(s_w^a)\psi_j^a \mathbf{K}^a \nabla \psi_i^a \cdot \mathbf{n}_e, p_c(s_w^a) - p_c(s_w^b) \right\rangle_e \\ &+ \frac{1}{2}\left\langle \lambda_n(s_w^a)\mathbf{K}^a \nabla \psi_i^a \cdot \mathbf{n}_e, p_c'(s_w^a)\psi_j^a \right\rangle_e + \frac{1}{2}\left\langle \lambda_t'(s_w^a)\psi_j^a \mathbf{K}^a \nabla \psi_i^a \cdot \mathbf{n}_e, p_w^a - p_w^b \right\rangle_e, \end{aligned} \tag{33}$$

$$\frac{\partial \hat{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_a}} = \gamma H\left(\|\lambda_t(s_w^k|_E)\mathbf{K}|_E\|_\infty\right)\left\langle p_c'(s_w^a)\psi_j^a, \psi_i^a\right\rangle_e. \tag{34}$$

2. Internal face integral for the derivative of $\mathcal{F}_{p,i}^{E_a}$ with respect to unknowns of the saturation on the neighboring element E_b:

$$\frac{\partial \mathcal{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} = \frac{\partial F_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} + \frac{\partial \tilde{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} + \frac{\partial \hat{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}}, \tag{35}$$

where

$$\begin{aligned}
\frac{\partial F_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} &= -\frac{1}{2}\left\langle \lambda_t'(s_w^b)\psi_j^b\mathbf{K}^b\nabla p_w^b\cdot\mathbf{n}_e, \psi_i^a\right\rangle_e - \frac{1}{2}\left\langle \lambda_n'(s_w^b)\psi_j^b\mathbf{K}^b p_c'(s_w^b)\nabla s_w^b\cdot\mathbf{n}_e, \psi_i^a\right\rangle_e \\
&- \frac{1}{2}\left\langle \lambda_n(s_w^b)\mathbf{K}^b p_c''(s_w^b)\psi_j^b\nabla s_w^b\cdot\mathbf{n}_e, \psi_i^a\right\rangle_e - \frac{1}{2}\left\langle \lambda_n(s_w^b)\mathbf{K}^b p_c'(s_w^b)\nabla\psi_j^b\cdot\mathbf{n}_e, \psi_i^a\right\rangle_e, \qquad (36)\\
\frac{\partial \tilde{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} &= -\frac{1}{2}\left\langle \lambda_n(s_w^a)\mathbf{K}^a\nabla\psi_i^a\cdot\mathbf{n}_e, p_c'(s_w^b)\psi_j^b\right\rangle_e, \qquad (37)\\
\frac{\partial \hat{F}_{p,i}^{E_a}}{\partial s_{w,j}^{E_b}} &= -\gamma H\left(\|\lambda_t(s_w^k|_E)\mathbf{K}|_E\|_\infty\right)\left\langle p_c'(s_w^b)\psi_j^b, \psi_i^a\right\rangle_e. \qquad (38)
\end{aligned}$$

The internal face integrals for the derivative of $\mathcal{F}_{p,i}^{E_b}$ with respect to unknowns of saturation on E_a and E_b can be derived in a similar way. Note that the element Jacobian J^E and the associated neighboring contribution of face integral are assembled into the global Jacobian matrix J_k using the point-block ordering in line with $\mathcal{F}(\mathbf{x}_k)$. We refer to [20] for more details for the construction of the global Jacobian matrix.

Remark 3.1. There are several techniques available to approximate the Jacobian with less programming effort, such as the multi-colored finite-difference (MCFD) method [14], the automatic differentiation (AD) method [26], and the Jacobian-free Newton Krylov (JFNK) method [34]. However, both MCFD and AD are in fact computationally more time-consuming and less accurate compared to the explicit hand-coded method [57]. On the other hand, we do not consider the JFNK method since the explicit form of the Jacobian is required to construct the preconditioner. Later in the paper, we present a performance comparison of the analytic method and the methods using an approximate Jacobian.

3.2. Additive Schwarz preconditioners

The global Jacobian constructed above is a large and sparse matrix that stems from the discretization of elliptic-hyperbolic PDEs with varying coefficients. Therefore, robust and efficient preconditioners are required to solve the linear Jacobian system (25). In this study, we employ the class of overlapping domain decomposition methods to build a preconditioner that is naturally suitable for large scale parallel processing.

We first rewrite (25) in the following general form

$$AM^{-1}y = b, \quad \text{with} \quad x = M^{-1}y, \tag{39}$$

where A is the Jacobian matrix, M is the preconditioner, x is the solution, and b is the right-hand side. Let n_p be the number of processor cores of the parallel computer. We first partition the grid into n_p subdomains using a graph-based partitioning approach (MeTis [31]), i.e., $\Omega_h = \Omega_1 \cup \cdots \cup \Omega_{n_p}$, where $\Omega_i \cap \Omega_j = \emptyset$, $\forall i \neq j$. The subvector associated with Ω_l is denoted as y_l. We then extend Ω_l to overlap with its neighbors by δ layers of elements and denote the overlapping subdomain as Ω_l^δ. On each overlapping subdomain, we define the corresponding subvector y_l^δ and the restriction operator R_l^δ that returns the vector of unknowns on Ω_l^δ, i.e.,

$$y_l^\delta = R_l^\delta y = (I \quad 0)\begin{pmatrix} y_l^\delta \\ y\backslash y_l^\delta \end{pmatrix}.$$

Fig. 1 shows a sample partition of a grid into 8 subdomains (left) and an example of an overlapping subdomain (right). We denote R_l^0 as the restriction operator that returns y_l defined on the nonoverlapping subdomain. Then, the one-level restricted additive Schwarz (RAS) preconditioner [10] is defined as

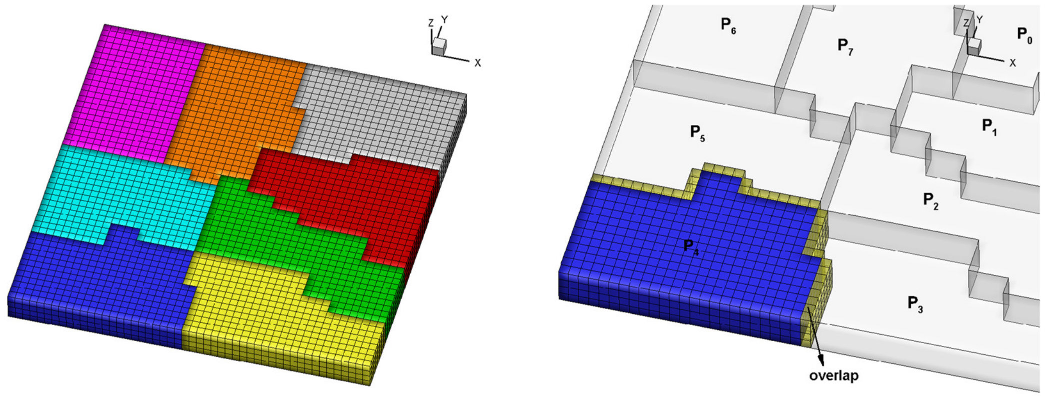


Fig. 1. (left) A sample partition of a uniform grid into 8 subdomains. (right) A non-overlapping subdomain colored in blue and one layer of overlap colored in yellow. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$M_{\text{one}}^{-1} = \sum_{l=1}^{n_p} \left(R_l^0\right)^T (A_l)^{-1} R_l^\delta, \tag{40}$$

$$A_l = R_l^\delta A \left(R_l^\delta\right)^T. \tag{41}$$

A larger overlap usually improves the numerical efficiency of the preconditioner in terms of the number of iterations, but also increases the communication time. A suitable size of overlap needs to be determined in order to achieve optimal performance in terms of the total compute time. The subdomain boundary condition is homogeneous Dirichlet for each variable. In (40), $(A_l)^{-1}$ is understood as the subdomain inverse, and its product with a vector is computed by solving a subdomain linear system inexactly. The choice of subdomain solver is important for the overall performance of the preconditioner on a specific computer with given amount of memory and cache. In our work, this is done by using a point-block incomplete LU factorization of A_l with fill-in level k (ILU(k)) based on the reverse Cuthill-McKee ordering. Using a larger fill-in level helps reduce the number of iterations, but this leads to an expensive solver in terms of the compute time and the memory usage. The impact of these factors will be discussed in the numerical tests.

Remark 3.2. In overlapping domain decomposition methods, some overlap is used for each subdomain, thus the boundary condition for the overlapping subdomain can be approximately chosen as the homogeneous Dirichlet boundary condition. This treatment is easy to implement, and has been adopted in many other non-elliptic multi-component equations such as the incompressible Navier-Stokes equations [40], the shallow water equations [57], and the compressible Euler equations [58].

To improve the scalability of the one-level RAS preconditioner (40) when using a large number of processor cores, we employ a hybrid preconditioner [52,57] by composing the one-level preconditioner $B_f = M_{\text{one}}^{-1}$ with a coarse-level preconditioner B_c in a multiplicative manner

$$M_{\text{two}}^{-1} = \text{Hybrid}(B_c, B_f) = B_c + B_f - B_f A B_c, \tag{42}$$

where $B_c = \mathcal{I}_c^f A_c^{-1} \mathcal{I}_f^c$, $\mathcal{I}_f^c$ and $\mathcal{I}_c^f$ are restriction and prolongation operators mapping between vectors defined on the fine level and the coarse level. Specifically, we first apply a coarse grid preconditioning

$$w = \left(\mathcal{I}_c^f A_c^{-1} \mathcal{I}_f^c\right) y, \tag{43}$$

and then correct the coarse solution by adding (without overlap) the fine level solution from each overlapping subdomain

$$x = w + \left(\sum_{l=1}^{n_p} \left(R_l^0\right)^T (A_l)^{-1} R_l^\delta\right) (y - Aw). \tag{44}$$

A schematic illustration of the hybrid two-level preconditioner is shown in Fig. 2.

The advantages of the two-level (or multi-level) preconditioner over the one-level preconditioner have been shown in other applications, such as thermal convection [51], fluid-structure interaction [6], flow control [59], and atmospheric dynamics [57]. On the coarse level, a smaller linear system associated with the Jacobian matrix A_c is solved by using GMRES with a one-level RAS preconditioner until a relative tolerance η_c is reached. Here A_c is obtained from the discretization of the equations on a coarse grid into which the fine grid can be nested. In such a case, the interpolation operators can be

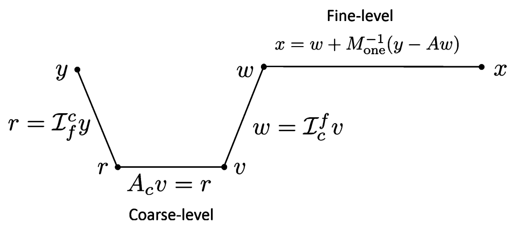


Fig. 2. A schematic illustration of the hybrid two-level Schwarz preconditioner. r and v are temporary vectors defined on the coarse level.

derived immediately and the routine to construct the Jacobian matrix can be reused. We use a flexible version of GMRES (FGMRES, see [49]) for solving the global Jacobian system since it is more suitable when the coarse problem is solved iteratively.

4. Numerical experiments

The algorithms are implemented using libMesh [32] for the finite element assembly, and PETSc [3] for the inexact Newton-Krylov solver. In our tests, the absolute permeability tensor is defined as $\mathbf{K} = k\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and k is a positive real number. Without loss of generality, the residual saturations in (8) are chosen to be zero, i.e. $s_e = s_w$, and β is fixed to 2. The wetting and non-wetting phases of the fluid are characterized with density $\rho_w = 1014$ kg/m^3 and $\rho_n = 859$ kg/m^3, respectively. The user-defined penalty parameter in the NIPG method is given as $\sigma = 10$ for all the tests. The relative tolerance for the nonlinear solver is set to be 10^{-6}. The relative tolerance for the linear solver is determined by η_k in (27). The choice of relative tolerance for the coarse linear solve η_c will be discussed in the test problems. The restart number of GMRES (or FGMRES) is 100. In the rest of this paper, 'NI' denotes the averaged number of nonlinear iterations per time step, 'LI' denotes the averaged number of linear iterations per nonlinear step, and 'Time' is the total compute time in second per time step, including the evaluation of Jacobian and residual.

For large-scale simulation on supercomputers with a large number of processor cores, the scalabilities of the algorithm with respect to the number of processor cores are of critical importance. In this paper, we focus on two scalability issues: (1) the strong scalability in terms of the total compute time when the size of the overall grid is fixed; and (2) the weak scalability in terms of the total compute time when the size of the grid per processor core is fixed.

4.1. Quarter five-spot problem

In this subsection, we validate the discretization scheme and the fully implicit solver using the classical quarter five-spot test case which consists of a symmetric flow pattern driven by two vertical wells on the diagonal of the domain [37]. The dimension of the domain is $[0, 250]$ m $\times$ $[0, 250]$ m $\times$ $[-20, 0]$ m and the media is initially filled by the non-wetting fluid. All boundaries are impermeable (Γ_0). Piecewise linear polynomials ($m = 1$) are used for the DG discretization on a uniform grid with $50 \times 50 \times 4$ cells in the respective directions. Each cell has 16 degrees of freedom (dofs) with respect to two variables (pressure and saturation) approximated by the linear combination of values on 8 nodes of the cell, leading to $160,000$ dofs in the global system. We set the viscosity to 1 cp for both fluid phases, giving a unit mobility ratio. A Peaceman well model [13] is used for the producer with radius 0.1 m and the bottom hole pressure is 10 bar. The parameter of capillary pressure is $B_c = 10$ and the gravity is ignored in this case. The time step size is $\Delta t = 0.2$ day. The one-level RAS method with the size of overlap $\delta = 1$ and the subdomain solver ILU(1) is used for preconditioning the linear Jacobian solver.

We first consider a homogeneous media with porosity $\phi = 0.2$ and permeability $k = 100$ mD. The wetting-phase fluid is injected with a flow rate 2.5×10^{-2} m^3/s at the injector. Fig. 3 shows the wetting-phase saturation at $t = 30$ days. Results are compared with those obtained from the Matlab Reservoir Simulation Toolbox (MRST [37]) using an IMPES method based on the cell center two-point flux-approximation (TPFA) scheme. Fig. 4 shows the comparison of the wetting-phase saturation profile along the diagonal at different times. Overall, a good agreement is observed for the results from the two different approaches.

We next consider a heterogeneous media with a Gaussian field of porosity and a lognormal distribution of permeability generated by MRST. The ranges of porosity and permeability are $[0.11, 0.4]$ and $[16.8, 281.6]$ mD respectively, as shown in Fig. 5(a)-(b). The wetting-phase fluid is injected with a flow rate 5×10^{-2} m^3/s at the injector. Other parameters are given the same as in the homogeneous case. Fig. 5(c)-(d) show the wetting-phase saturation at $t = 10$ days. It is seen that the proposed implicit DG method successfully resolves the non-uniform evolution of the saturation, and the result is consistent with that obtained using MRST.

4.2. Displacement problem in a regular domain

When a displacement front propagates through a porous medium, the combination of viscosity differences and permeability heterogeneity may introduce viscous fingering effects [37]. In this subsection, we consider the displacement of a

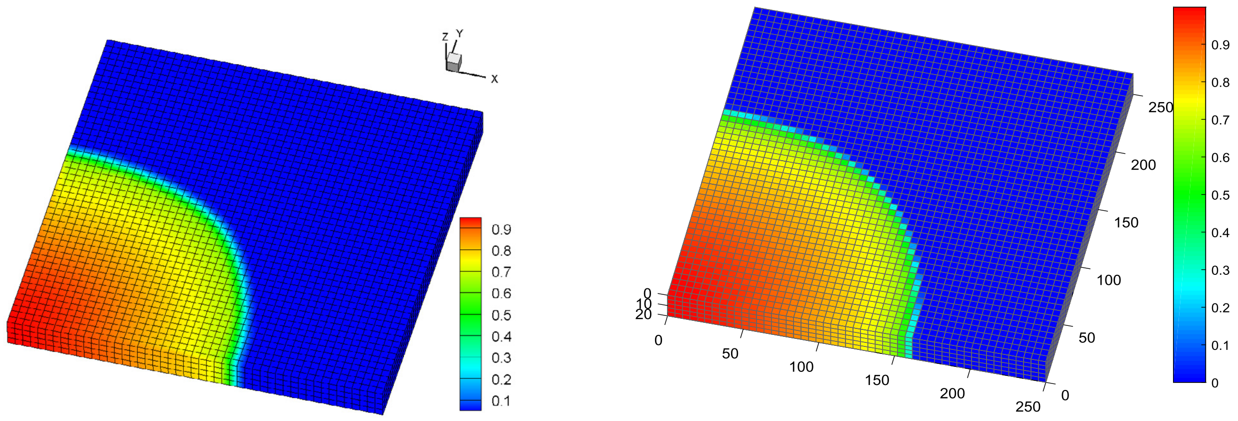


Fig. 3. Wetting-phase saturation at 30 days for the quarter five-spot problem with homogeneous media. (left) Result obtained using the presented implicit DG method, (right) result obtained using MRST.

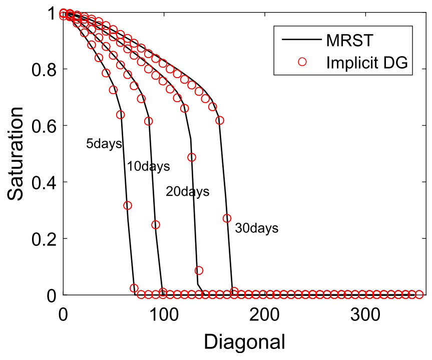
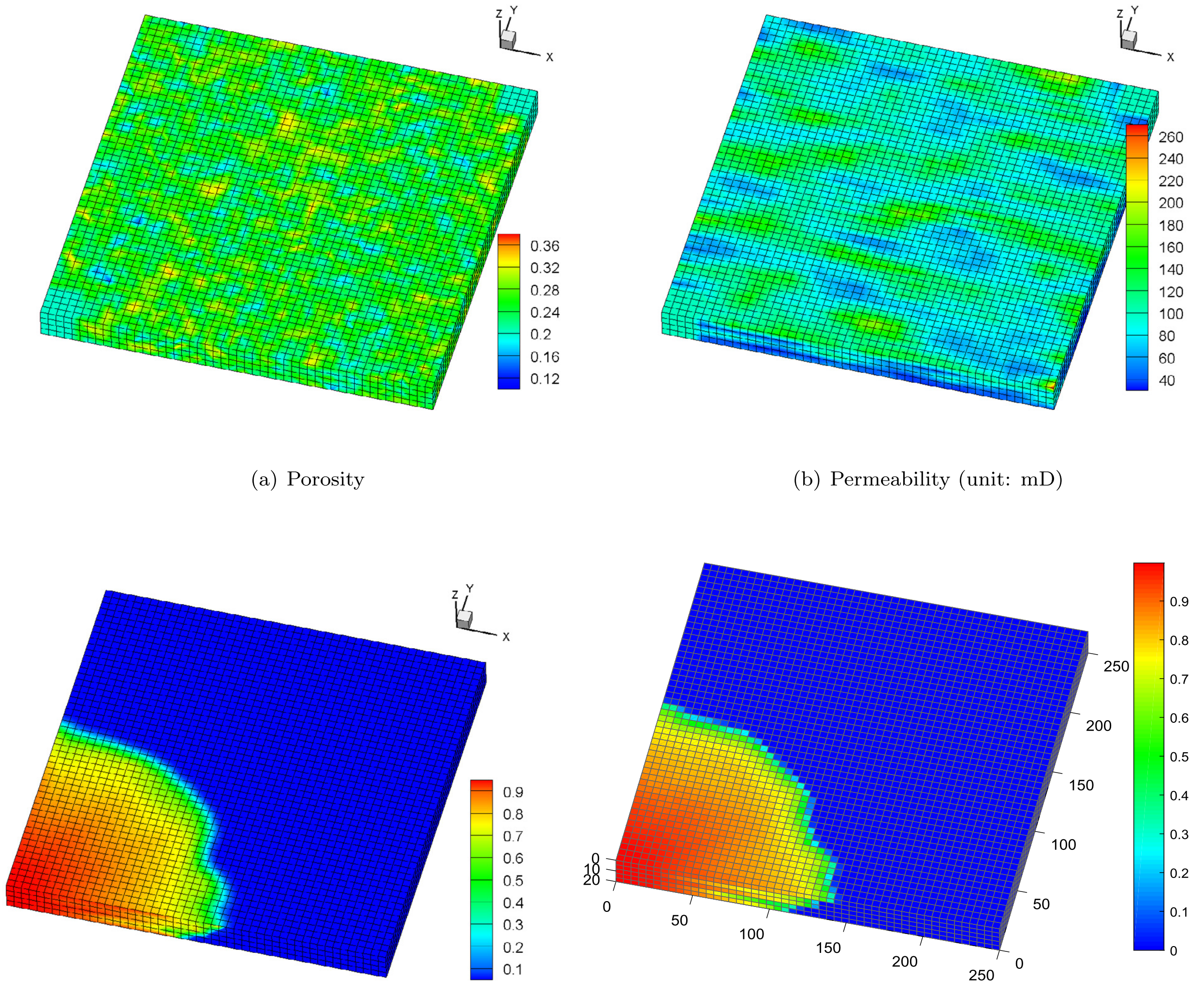


Fig. 4. Comparison of the diagonal profile of wetting-phase saturation at different times for the quarter five-spot problem with homogeneous media.

viscous fluid by injecting a less viscous fluid in a heterogeneous media, and study the impact of some parameters on the performance of our solver. The dimension of the domain is $[0, 80]$ m$\times[0, 400]$ m$\times[-160, 0]$ m divided into $10 \times 50 \times 20$ cells. The plane $y = 0$ m is treated as the inlet boundary (Γ_{in}), while the plane $y = 400$ m is treated as the outlet boundary (Γ_{out}). The other boundaries are impermeable (Γ_0). The permeability field is generated by MRST with a lognormal distribution in the range of $[7.0, 453.3]$ mD and the porosity has a Gaussian distribution with the range of $[0.05, 0.5]$, as shown in Fig. 6(a)-(b). The media is initially filled by the non-wetting phase with viscosity 3 cp at $y > 20$ m and the wetting phase with viscosity 1 cp at $y \le 20$ m. The injected rate of wetting phase at the inlet is $\int_{\Gamma_{in}} f_w^{in} \mathrm{d}s = 2000$ m^3/day. The pressure value at the outlet is $p_w^{out} = 5$ bar. The parameter of capillary pressure is $B_c = 10$ and the gravitational acceleration constant is $g = 9.81$ m/s^2. The time step size is $\Delta t = 0.1$ day. Piecewise quadratic polynomials ($m = 2$) are used for the DG discretization, leading to 27 dofs for each variable in each cell. The global system has totally 540,000 dofs. The one-level RAS method with the size of overlap $\delta = 1$ and the subdomain solver ILU(1) is used for preconditioning the linear Jacobian solver. The simulations are carried out using 256 processor cores. The wetting-phase saturation at 80 days and 240 days are shown in Fig. 6(c)-(d). It is observed that the displacing phase forms a weak shock front that 'fingers' rapidly through the path of highest permeability.

It is well known that the capillary pressure parameter B_c has a great impact on the flow behavior in porous media. In Table 1, we show the performance of the proposed solver using different values of B_c. From the table we see that, if the time step size Δt is small, the nonlinear solver is robust with respect to the change of B_c. On the other hand, when a larger Δt is used, the number of nonlinear iterations increases as B_c increases, and the solver consumes more compute time during each time step, due to the stronger nonlinearity induced by the capillary pressure.

We next investigate the impact of two important factors on the parallel performance of the one-level RAS preconditioner: the fill-in level k of the ILU subsolve and the size of overlap δ. We fix the grid size to $10 \times 50 \times 20$ and vary the number of processor cores n_p from 32 to 256. The time step size is $\Delta t = 0.1$ day. Results are summarized in Table 2. The numbers reported in the table are obtained by taking average for 10 time steps. The table clearly indicates that the number of nonlinear iterations is almost independent of n_p, k, and δ, while the number of linear iterations increases as n_p increases. In general, increasing the fill-in level or the size of overlap results in a stronger linear preconditioner, thus the number of linear iterations decreases. However, this may not result in a better performance in terms of the total compute time, since



(a) Porosity (b) Permeability (unit: mD)

(c) Implicit DG, 10 days (d) MRST, 10 days

Fig. 5. Quarter five-spot problem with heterogeneous media. (c) Result obtained using the presented implicity DG method, (d) result obtained using MRST.

the cost per iteration for communication and computation increases. It is found that the choice of $\delta = 1$ and ILU(1) leads to the smallest amount of total compute time and the best parallel efficiency among the tested cases.

Due to the complicated operations involved in the DG discretization, the calculation of the Jacobian is expensive. The situation becomes worse when higher order polynomials are used since the resulting Jacobian is much denser. To demonstrate the benefits using the analytic Jacobian, we compare the performance with the MCFD method and the JFNK method offered by PETSc [3]. In JFNK, the analytically calculated matrix is used for preconditioning the Jacobian system to guarantee the convergence of the linear solver. The results are shown in Table 3, where the grid size is $5 \times 25 \times 10$ and the time step size is $\Delta t = 6.25 \times 10^{-3}$ day. It is clear that the analytic method keeps the number of nonlinear iterations stable for the test cases with order $m = 1$ and $m = 2$, and its performance is far better than the methods using an approximate Jacobian in terms of the total compute time.

4.3. Displacement problem in a complex domain

In this subsection, the tested media has a complex geometry with topological change on the top surface. For comparison, we use the same boundary conditions and the same heterogeneous fields of porosity and permeability for the porous medium as in the regular domain case, see Fig. 7(a)-(b). The simulation of two-phase flow in this complicated media requires a fine enough unstructured grid to resolve the dynamic process of the saturation, which causes complication in the numerical modeling. The one-level preconditioner is not completely satisfactory in this case, therefore, we employ the two-

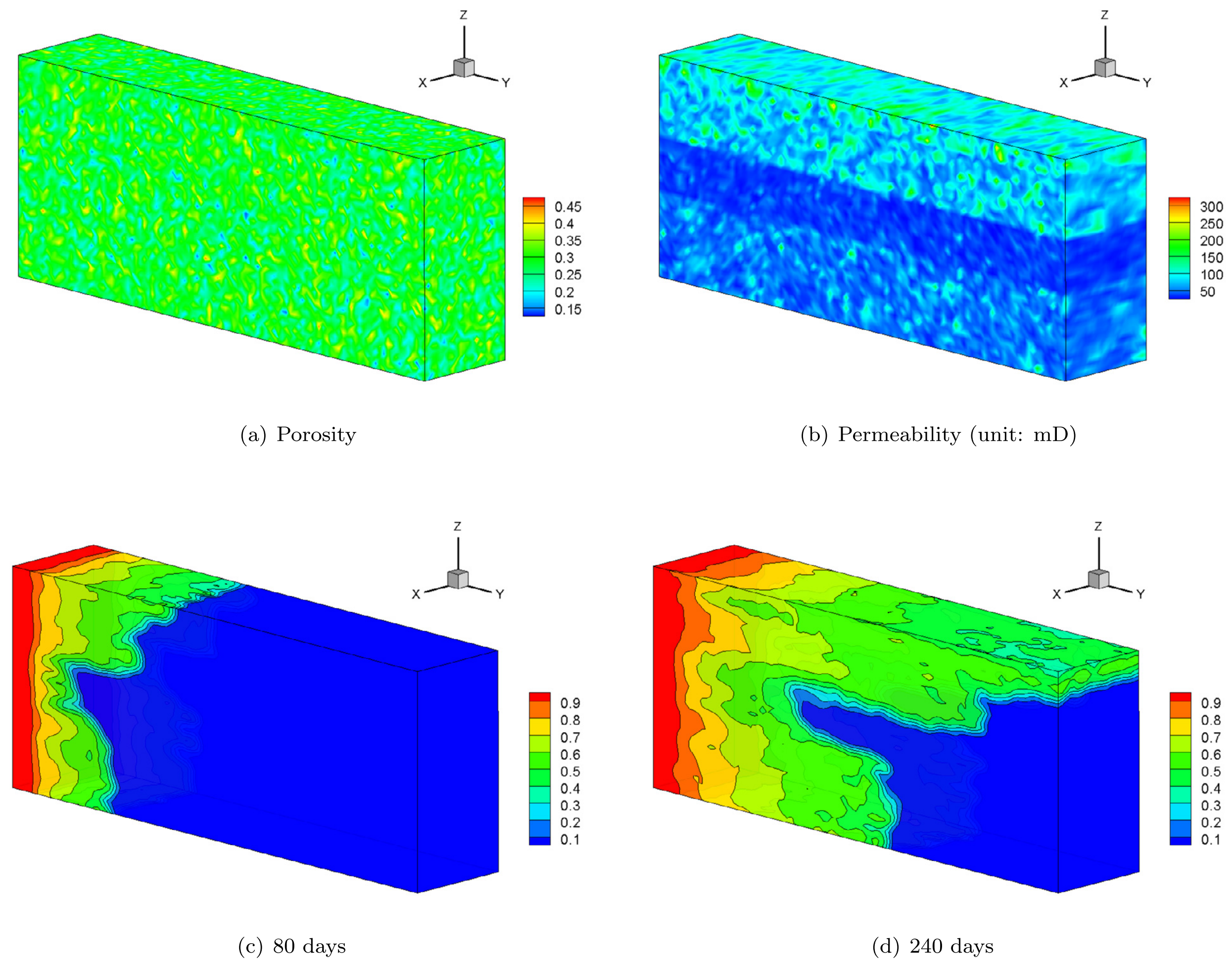


Fig. 6. Displacement of a viscous fluid by injecting a less viscous fluid in a regular domain with heterogeneous media.

Table 1
Performance of the proposed solver with respect to different values of B_c. The grid size is $10 \times 50 \times 20$. The one-level RAS preconditioner is used with the size of overlap $\delta = 1$ and the subdomain solver is ILU(1). The simulations are carried out using 256 processor cores. The mark '–' indicates that the case fails to converge.

B_c	$\Delta t = 0.05$ day			$\Delta t = 0.1$ day			$\Delta t = 0.2$ day		
	NI	LI	Time	NI	LI	Time	NI	LI	Time
0	4	46.2	8.15	3.9	51.0	8.18	4	56.8	8.62
20	4	48.5	8.24	4.1	45.1	8.30	6.8	34.7	14.23
60	4	47.1	8.16	5.5	45.6	12.66	–	–	–

level preconditioner to improve the convergence and scalability of the overall solver. Piecewise quadratic polynomials are used for the DG discretization on an unstructured grid with 72,000 elements and 602,905 nodes, resulting in 3,888,000 dofs. The coarse grid has 9,000 elements and 78,813 nodes, resulting in 486,000 dofs. The fine and coarse grids with a sample partition into 16 subdomains are shown in Fig. 8. In the two-level RAS preconditioner, the size of overlap on the fine level δ_f and that on the coarse level δ_c are both set to 1. We use ILU(3) as the subdomain solver for both levels. The GMRES iteration on the coarse level is stopped upon reaching the relative tolerance $\eta_c = 0.1$ or a maximum iteration count of 100. The time step size is $\Delta t = 0.05$ day. The simulation is carried out using 1,024 processor cores. Other parameters of the problem are the same as in the regular domain case. Results are plotted in 7(c)-(d). The overall displacement process is similar to the regular domain case. One distinct feature is that, when the displacement front passes through the top layer, a portion of the resident fluid is trapped in the corners and bumps near the top surface. This indicates that the flow behavior is influenced by the topological variation of the domain.

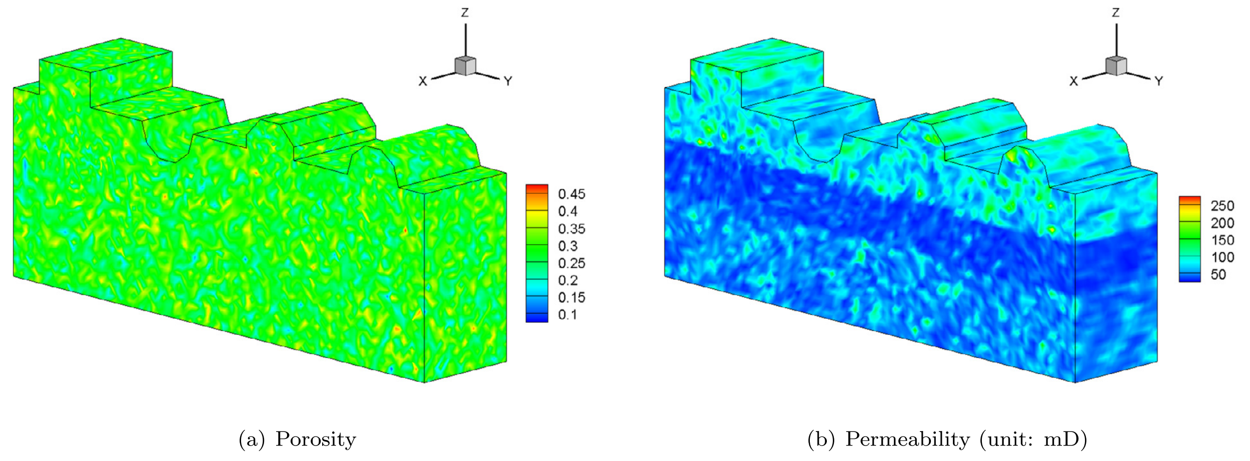
Table 2
Performance of the proposed solver with respect to different fill-in levels of the ILU subsolve and different sizes of overlap in the one-level RAS preconditioner. The grid size is $10 \times 50 \times 20$. The time step size is $\Delta t = 0.1$ day.

The size of overlap $\delta = 1$									
n_p	ILU(1)			ILU(2)			ILU(3)		
	NI	LI	Time	NI	LI	Time	NI	LI	Time
32	3.9	29.6	42.62	3.7	15.1	48.32	3.6	12.2	64.24
64	3.9	37.6	22.65	3.9	21.0	28.70	3.9	19.8	36.54
128	3.9	38.3	14.26	3.9	30.2	17.21	3.9	29.2	22.31
256	3.9	46.9	8.00	3.9	33.5	8.97	3.9	38.2	11.24
The size of overlap $\delta = 2$									
n_p	ILU(1)			ILU(2)			ILU(3)		
	NI	LI	Time	NI	LI	Time	NI	LI	Time
32	3.9	25.6	47.74	3.6	11.6	56.15	3.7	9.7	87.36
64	3.8	30.9	36.05	3.6	15.7	42.13	3.9	11.9	64.45
128	3.8	29.3	17.22	3.8	18.1	22.99	3.9	13.9	36.05
256	3.9	29.5	10.11	3.7	18.0	12.67	3.9	16.9	20.18

Table 3
Comparison of performance using MCFD, JFNK*, and analytic methods for the construction of the Jacobian. The grid size is $5 \times 25 \times 10$ and the time step size is $\Delta t = 6.25 \times 10^{-3}$ day. The size of overlap in RAS is $\delta = 1$ and the subdomain solver is ILU(1). JFNK* means using the Jacobian-free method for matrix-vector multiplication, while using the analytically calculated matrix for preconditioning the Jacobian system.

Order	n_p	MCFD			JFNK*			Analytic		
		NI	LI	Time	NI	LI	Time	NI	LI	Time
$m = 1$	64	4.7	13.5	9.17	3.8	13.5	0.74	3	15.2	0.138
	128	4.7	16.7	6.33	3.9	16.8	0.57	3	21.4	0.095
	256	4.7	19.4	3.80	5.9	17.5	0.58	3	32.9	0.077
$m = 2$	64	7.9	10.1	657.77	4.2	14.3	12.59	3	18.4	2.43
	128	8.1	10.8	436.96	4	18.1	9.15	3	21.0	1.53
	256	8.1	14.8	218.39	5.2	21.3	7.41	3	32.2	0.85

It is worthy to discuss how the coarse solution affects the performance of the two-level preconditioner. In practice, a better coarse solution would help reduce the total number of iterations, but the overall compute time may increase. We are interested in two factors that determine the accuracy of the coarse solution: the size of the coarse grid and the stopping condition for the coarse solve η_c. Table 4 shows the performance using the one-level preconditioner and the two-level preconditioner with different coarse grids and different η_c. The numbers reported in the table are obtained by taking average for 5 time steps. The two coarse grids are obtained by coarsening the fine grid uniformly by one or two ratios of 2. The time step size is $\Delta t = 0.05$ day. In the table, 'dof$_c$' denotes the degrees of freedom on the coarse level; 'LI$_c$' denotes the average number of linear iterations for the coarse solve per global linear step; 'Time$_c$' denotes the time spent on the coarse solve; 'Time$_{\mathcal{I}_f^c}$' denotes the time spent on the restriction operator $\mathcal{I}_f^c$, and 'Time$_{\mathcal{I}_c^f}$' denotes the time spent on the prolongation operator $\mathcal{I}_c^f$. Fig. 9 shows the total compute time for the tested cases. From the table and the figure, the following observations can be made: (1) When a small number of processor cores is used, i.e. $n_p = 128$, the one-level preconditioner requires less compute time than the two-level preconditioner. However, as n_p increases, it takes more linear iterations to converge and the number of nonlinear iterations grows, leading to a poor scalability. (2) The two-level preconditioner greatly reduces the numbers of linear iterations while keeping the numbers of nonlinear iterations stable, yielding a good numerical convergence. (3) On the coarse level, using a relatively fine grid leads to more compute time on the coarse solve. Alternatively, more time is spent on the fine level and the prolongation if a coarser grid is used. For a clearer comparison, the times spent on different levels are provided in Fig. 10. (4) When a smaller η_c is used, the coarse solve requires more iterations and more compute time to meet the stopping condition, but the reduction of the number of global linear iterations is limited. (5) To make the two-level preconditioner scalable, the compute times for the coarse solve, the restriction, and the prolongation should also be scalable. Overall, among the tested cases, the two-level preconditioner with a relatively fine grid and $\eta_c = 0.1$ for the coarse level proves to be the best choice in terms of total compute time and parallel scalability. Shadid et al. [51] reached a similar conclusion for a 3D thermal convection simulation. They compared three domain-decomposition preconditioners: one-level, two-level with an exact coarse grid solve, and two-level with an iterative approximate coarse grid solve. While the two-level exact coarse grid solve has the best scaling of iterations per Newton step, the two-level approximate coarse grid solve is the best in execution time.



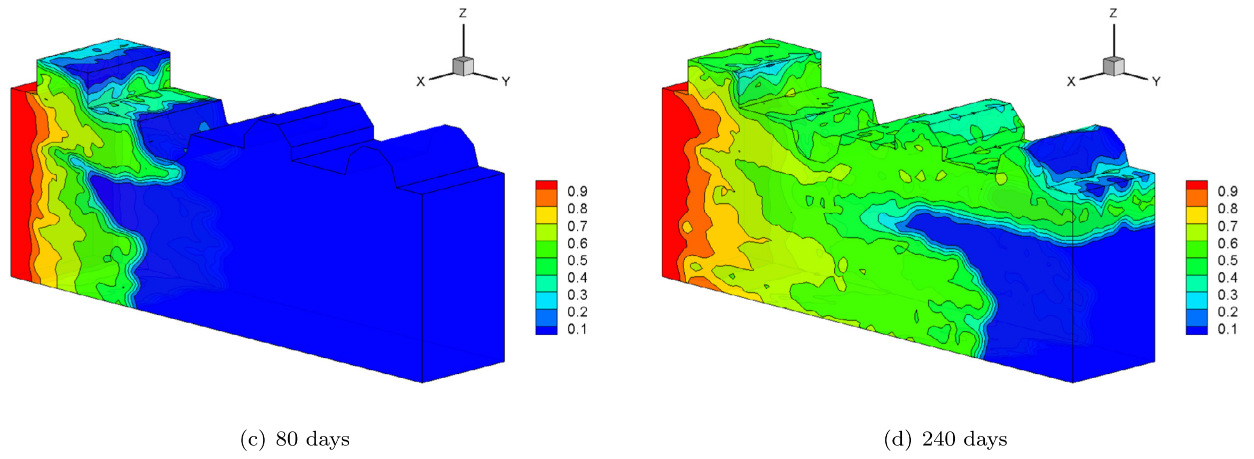


Fig. 7. Displacement of a viscous fluid by injecting a less viscous fluid in a complex domain with heterogeneous media.

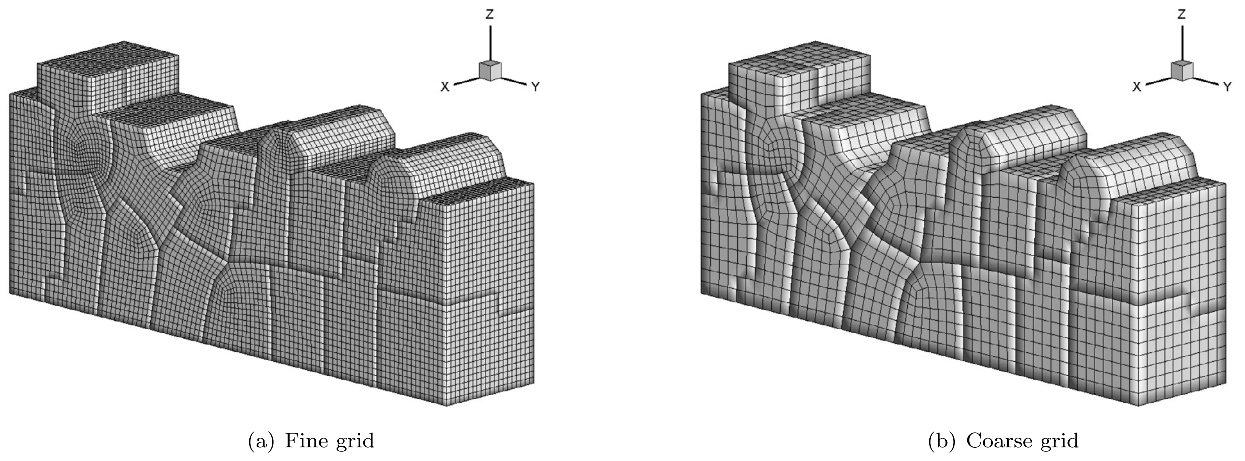


Fig. 8. The fine and coarse grids with a sample partition into 16 subdomains for the complex domain case. The fine grid has 72, 000 elements and 602, 905 nodes, resulting in 3, 888, 000 dofs. The coarse grid has 9, 000 elements and 78, 813 nodes, resulting in 486, 000 dofs.

Table 4
Performance of the proposed solver using the two different preconditioners. The fine grid has 72, 000 elements and 602, 905 nodes, resulting in 3, 888, 000 dofs. The size of overlap is 1 and the subdomain solver is ILU(3) for both preconditioners. The time step size is $\Delta t = 0.05$ day.

Level	n_p	dof$_c$	η_c	NI	LI	LI$_c$	Time$_c$	Time$_{\mathcal{I}_f^c}$	Time$_{\mathcal{I}_c^f}$	Time
One	128			6.4	23.6					312.90
One	256			6.4	26.8					177.06
One	512			6.6	36.6					102.07
One	1,024			7.8	74.4					86.46
Two	128	6.075×10^4	0.1	6.2	12.5	17.4	6.91	1.97	47.37	353.84
Two	256	6.075×10^4	0.1	6.0	10.8	23.3	4.09	0.98	22.12	186.16
Two	512	6.075×10^4	0.1	6.6	10.1	26.1	1.92	0.69	13.91	109.53
Two	1,024	6.075×10^4	0.1	6.6	12.5	42.5	2.37	0.67	11.46	73.78
Two	128	4.86×10^5	0.1	5.6	12.9	18.6	58.67	1.94	30.44	355.68
Two	256	4.86×10^5	0.1	5.6	13.0	25.0	32.73	1.15	17.73	174.79
Two	512	4.86×10^5	0.1	5.6	13.8	31.1	19.13	0.80	11.41	99.81
Two	1,024	4.86×10^5	0.1	5.0	12.8	38.3	12.48	0.53	6.34	59.57
Two	128	4.86×10^5	0.01	7.0	10.3	36.4	181.38	3.41	53.09	459.20
Two	256	4.86×10^5	0.01	5.6	11.0	49.3	118.97	1.85	27.27	226.18
Two	512	4.86×10^5	0.01	5.0	10.8	62.3	60.20	0.94	13.13	111.97
Two	1,024	4.86×10^5	0.01	5.6	11.4	60.8	46.33	0.94	10.70	81.20

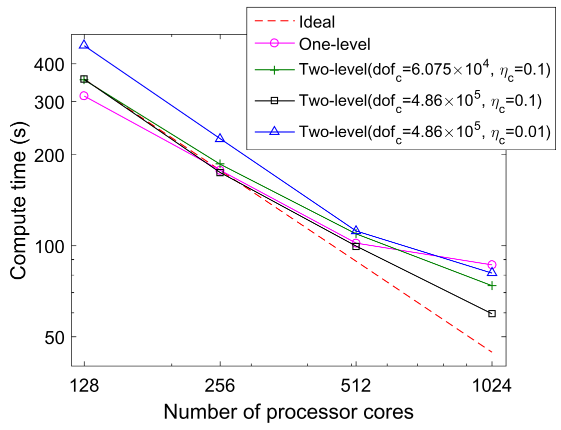


Fig. 9. Total compute times of the proposed solver using the one-level preconditioner and the two-level preconditioner with different coarse grids and different relative tolerance for the coarse solve. The figure is drawn on a log-log scale.

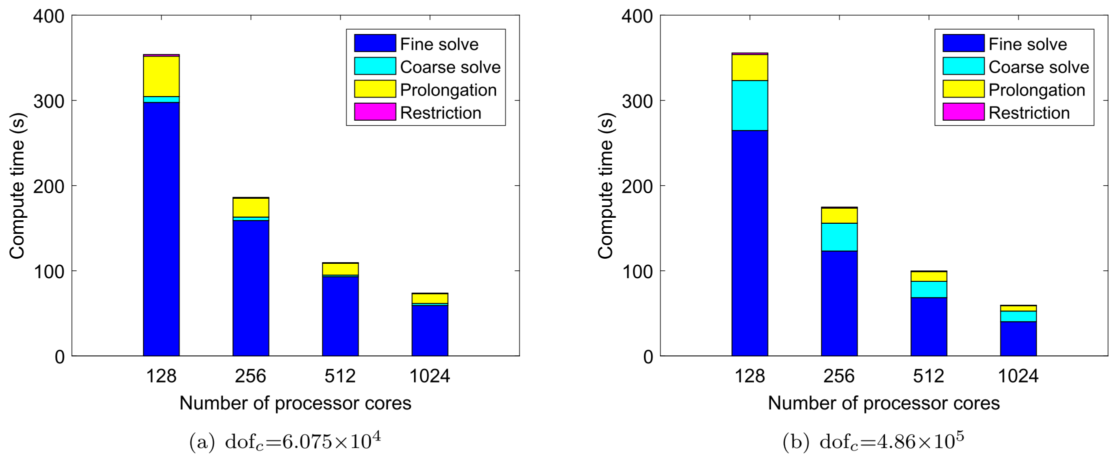


Fig. 10. Compute times on different levels of the proposed solver using the two-level preconditioner with different coarse grids. The relative tolerance for the coarse solve is $\eta_c = 0.1$ for both cases.

Table 5
Strong scalability test for the proposed solver using the one-level and two-level preconditioners. The fine grid has 168, 584 elements and 1, 396, 073 nodes, resulting in 9, 103, 536 dofs. The coarse grid has 21, 073 elements and 180, 549 nodes, resulting in 1, 137, 942 dofs. The relative tolerance for the coarse solve is $\eta_c = 0.1$. ILU(3) is used as the subdomain solver for both preconditioners.

Level	Δt (day)	Overlap	n_p	NI	LI	Time	Speedup
One	6.25×10^{-3}	$\delta = 2$	256	3.4	13.8	218.09	1
One	6.25×10^{-3}	$\delta = 2$	512	3.6	18.2	149.99	1.45
One	6.25×10^{-3}	$\delta = 2$	1,024	3.4	17.5	79.42	2.75
One	6.25×10^{-3}	$\delta = 2$	2,048	3.6	23.9	61.03	3.57
One	6.25×10^{-3}	$\delta = 2$	4,096	3.6	25.9	28.21	7.73
One	6.25×10^{-3}	$\delta = 2$	8,192	3.6	37.9	16.94	12.87
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	256	3	9.7	237.29	1
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	512	3	9.9	118.16	2.01
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	1,024	3	9.8	59.91	3.96
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	2,048	3	13.3	39.02	6.08
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	4,096	3	12.2	20.67	11.48
Two	6.25×10^{-3}	$\delta_f = \delta_c = 1$	8,192	3	9.2	8.69	27.31
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	256	3.2	10.1	279.96	1
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	512	3.2	9.4	133.73	2.09
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	1,024	3.2	8.5	68.96	4.06
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	2,048	3.2	13.5	49.75	5.63
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	4,096	3.2	10.0	27.68	10.11
Two	2.5×10^{-2}	$\delta_f = 1, \delta_c = 2$	8,192	3.2	9.1	15.11	18.53

We next examine the strong scalability of the proposed solver on a large number of processor cores, and analyze the robustness of the one-level and two-level preconditioners using different time step sizes. In the tests, the fine grid has 168, 584 elements and 1, 396, 073 nodes, resulting in 9, 103, 536 dofs. The coarse grid has 21, 073 elements and 180, 549 nodes, resulting in 1, 137, 942 dofs. In the two-level RAS preconditioner, the relative tolerance for the coarse solve is $\eta_c = 0.1$. ILU(3) is used as the subdomain solver for both preconditioners. Results of the strong scalability test on different numbers of processor cores are reported in Table 5. We remark that firstly, the one-level preconditioner fails to converge when using $\delta = 1$, while the two-level preconditioner is less sensitive to the overlapping size. Secondly, when we increase the time step size from 6.25×10^{-3} day to 2.5×10^{-2} day, the one-level preconditioner fails even with a larger size of overlap together with the exact LU factorization as the subdomain solver. On the other hand, the two-level preconditioner performs well with this time step size on a large number of processor cores. Fig. 11 (top) shows the total compute times of the proposed solver using the one-level and two-level preconditioners with $\Delta t = 6.25 \times 10^{-3}$ day, where 'sp$_a$' denotes the speedup of the algorithm using the two-level preconditioner over the one-level preconditioner. We can see from the figure that the two-level preconditioner is faster than the one-level preconditioner when n_p is greater than 256, and the speedup reaches 1.95 when n_p is up to 8,192. Fig. 11 (bottom) shows the speedup of the proposed solver with respect to the number of processor cores. As n_p increases from 256 to 8, 192, the two-level preconditioner results in a better performance than the one-level preconditioner for both time step sizes.

We further test our algorithm in terms of weak scalability in which we refine the grid as n_p increases so that almost the same number of unknowns per processor core is maintained. The time step size is fixed at $\Delta t = 6.25 \times 10^{-3}$ day. The other parameters are the same with the strong scalability test. Table 6 shows the weak scaling results of the proposed solver. As shown in the table, increasing the grid size barely affects the number of nonlinear iterations for both preconditioners, while the number of linear iterations for the one-level method increases rapidly. Adding the coarse level does reduce the number of linear iterations and the total compute time. Fig. 12 shows the performance of the proposed solver using the one-level and two-level preconditioners. It is seen that the two-level method results in a flatter curve than the one-level method, and the compute time increases only by 53% as n_p increases from 256 to 8,192 (32 times larger).

5. Concluding remarks

We developed a scalable fully implicit solver for incompressible two-phase flows in porous media based on the framework of overlapping domain decomposition methods on 3D unstructured grids. The governing equations are discretized by a backward Euler scheme in time and a discontinuous Garlerkin finite element method in space. The resulting nonlinear system is solved by the class of parallel Newton-Krylov-Schwarz algorithms with analytic Jacobian. We tested the algorithm using several 3D heterogeneous medias on unstructured grids, and compared the performance between the one-level and two-level additive Schwarz preconditioners in terms of the number of iterations and the total compute time. Results of numerical experiments show that the proposed solver with the two-level preconditioner is more robust and scalable for problems with millions of unknowns on a supercomputer with more than 8, 000 processor cores.

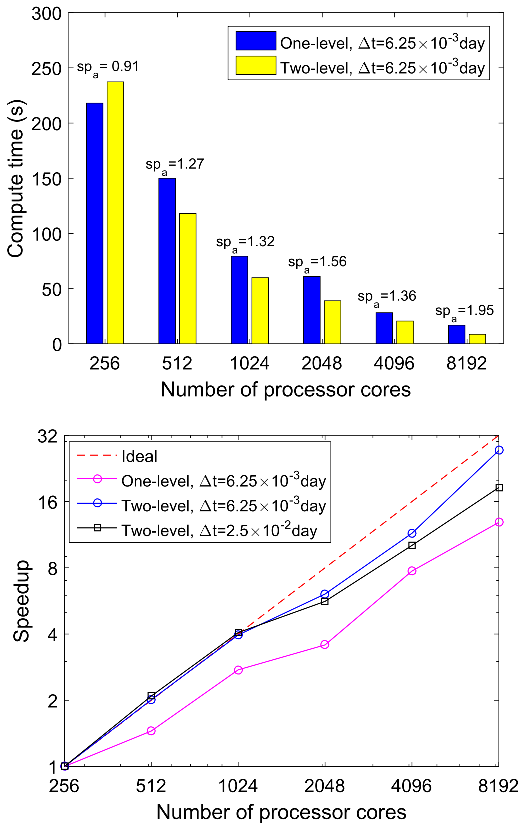


Fig. 11. Strong scalability test: (top) Total compute times of the proposed solver using the one-level preconditioner and the two-level preconditioner with $\Delta t = 6.25 \times 10^{-3}$ day. 'sp$_a$' denotes the speedup of the algorithm using the two-level preconditioner over the one-level preconditioner. (bottom) Speedup of the proposed solver with respect to the number of processor cores. The bottom panel is drawn on a log-log scale.

Table 6
Weak scalability test for the proposed solver using the one-level and two-level preconditioners. The time step size is $\Delta t = 6.25 \times 10^{-3}$ day. The relative tolerance for the coarse solve is $\eta_c = 0.1$. ILU(3) is used as the subdomain solver for both preconditioners. 'dof$_f$' and 'dof$_c$' denote the degrees of freedom on the fine and coarse level, respectively.

Level	dof$_f$	dof$_c$	Overlap	n_p	NI	LI	Time
One	245,376		$\delta = 2$	256	3	16.7	8.23
One	509,760		$\delta = 2$	512	3	20.5	9.17
One	984,960		$\delta = 2$	1,024	3	23.3	9.55
One	2,115,072		$\delta = 2$	2,048	3	30.2	10.45
One	4,652,208		$\delta = 2$	4,096	3	29.8	11.28
One	9,103,536		$\delta = 2$	8,192	3.6	37.9	16.94
Two	245,376	30,672	$\delta_f = \delta_c = 1$	256	3	9.6	5.68
Two	509,760	63,720	$\delta_f = \delta_c = 1$	512	3	9.3	6.05
Two	984,960	123,120	$\delta_f = \delta_c = 1$	1,024	3	10.0	6.45
Two	2,115,072	264,384	$\delta_f = \delta_c = 1$	2,048	3	8.7	6.74
Two	4,652,208	581,526	$\delta_f = \delta_c = 1$	4,096	3	9.3	8.25
Two	9,103,536	1,137,942	$\delta_f = \delta_c = 1$	8,192	3	9.2	8.69

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The first author was supported in part by the National Natural Science Foundation of China (11701547), the second author was supported in part by the National Natural Science Foundation of China (11901296) and by the Natural Science

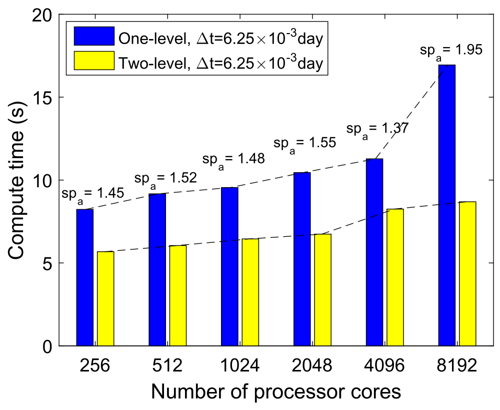


Fig. 12. Weak scalability test: Total compute times of the proposed solver using the one-level preconditioner and the two-level preconditioner with $\Delta t = 6.25 \times 10^{-3}$ day. 'sp_a' denotes the speedup of the algorithm using the two-level preconditioner over the one-level preconditioner.

Foundation for Young Scientists of Jiangsu (BK20180450). This research was also supported by the Extreme Computing Research Center of the King Abdullah University of Science and Technology.

References

[1] T. Arbogast, M. Juntunen, J. Pool, M.F. Wheeler, A discontinuous Galerkin method for two-phase flow in a porous medium enforcing $H(div)$ velocity and continuous capillary pressure, Comput. Geotech. 17 (2013) 1055–1078.
[2] A. Baggag, H. Atkins, D.E. Keyes, Parallel implementation of the discontinuous Galerkin method, in: Proceedings of Parallel CFD'99, 1999, pp. 115–122.
[3] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc Users Manual, Argonne National Laboratory, 2019.
[4] P. Bastian, B. Rivière, Discontinuous Galerkin methods for two-phase flow in porous media, Tech. Rep. 2004-28, IWR, University of Heidelberg, 2004.
[5] P. Bastian, A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure, Comput. Geosci. 18 (2014) 779–796.
[6] A.T. Barker, X.-C. Cai, Two-level Newton and hybrid Schwarz preconditioners for fluid-structure interaction, SIAM J. Sci. Comput. 32 (2010) 2390–2417.
[7] A.T. Barker, S.C. Brenner, L.-Y. Sung, Overlapping Schwarz domain decomposition preconditioners for the local discontinuous Galerkin method for elliptic problems, J. Numer. Math. 19 (2011) 165–187.
[8] M. Bernacki, L. Fezoui, S. Lanteri, S. Piperno, Parallel discontinuous Galerkin unstructured mesh solvers for the calculation of three-dimensional wave propagation problems, Appl. Math. Model. 30 (2006) 744–763.
[9] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, D.P. Young, Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation, SIAM J. Sci. Comput. 192 (1998) 46–65.
[10] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (1999) 792–797.
[11] Z. Chen, R. Ewing, From single-phase to compositional flow: applicability of mixed finite elements, Transp. Porous Media 57 (1997) 225–242.
[12] Z. Chen, G. Huan, B. Li, An improved IMPES method for two-phase flow in porous media, Transp. Porous Media 54 (2004) 361–376.
[13] Z. Chen, G. Huan, Y. Ma, Computational Methods for Multiphase Flows in Porous Media, SIAM, Philadelphia, PA, USA, 2006.
[14] T.F. Coleman, J.J. Moré, Estimation of sparse Jacobian matrices and graph coloring problems, SIAM J. Numer. Anal. 20 (1983) 187–209.
[15] M. Cusini, A.A. Lukyanov, J. Natvig, H. Hajibeygi, Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media, J. Comput. Phys. 299 (2015) 472–486.
[16] C.N. Dawson, H. Klíe, M.F. Wheeler, C.S. Woodward, A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver, Comput. Geosci. 1 (1997) 215–249.
[17] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, Philadelphia, 1996.
[18] A.H. Dogru, L.S.K. Fung, U. Middya, T.M. Al-Shaalan, J.A. Pita, K. HemanthKumar, H.J. Su, J.C.T. Tan, H. Hoy, W.T. Dreiman, W.A. Hahn, R. Al-Harbi, A. Al-Youbi, N.M. Al-Zamel, M. Mezghani, T. Al-Mani, A next generation parallel reservoir simulator for giant reservoirs, in: SPE 119272 Presented at the SPE Reservoir Simulation Symposium, Texas, Feb. 2-4, 2009.
[19] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[20] Y. Epshteyn, B. Rivière, Fully implicit discontinuous finite element methods for two-phase flow, Appl. Numer. Math. 57 (2007) 383–401.
[21] Y. Epshteyn, B. Rivière, Analysis of *hp* discontinuous Galerkin methods for incompressible two-phase flow, J. Comput. Appl. Math. 225 (2009) 487–509.
[22] A. Ern, I. Mozolevski, L. Schuh, Discontinuous Galerkin approximation of two-phase flows in heterogeneous porous media with discontinuous capillary pressures, Comput. Methods Appl. Mech. Eng. 199 (2010) 1491–1501.
[23] O. Eslinger, Discontinuous Galerkin finite element methods applied to two-phase, air-water flow problems, Ph.D. thesis, University of Texas at Austin, 2005.
[24] R. Ewing, R. Heinemann, Incorporation of mixed finite element methods in compositional simulation for reduction of numerical dispersion, in: Reservoir Simulation Symposium, No. SPE12267, 1983.
[25] S. Gries, K. Stüben, G.L. Brown, D. Chen, D.A. Collins, Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations, SPE J. 19 (2014) 726–736.
[26] A. Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, SIAM, Philadelphia, 2000.
[27] W.D. Gropp, D.E. Keyes, L. McInnes, M. Tidriri, Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD, Int. J. High Perform. Comput. Appl. 14 (2000) 102–136.
[28] D. Gunasekera, P. Childs, J. Herring, J. Cox, A multi-point flux discretization scheme for general polyhedral grids, in: International Oil and Gas Conference and Exhibition, SPE48855, 1998.
[29] H. Hoteit, A. Firoozabadi, Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures, Adv. Water Resour. 31 (2008) 56–73.

[30] J. Hou, J. Chen, S. Sun, Z. Chen, Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media, J. Comput. Appl. Math. 307 (2016) 262–283.
[31] G. Karypis, V. Kumar, MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0, University of Minnesota, Minneapolis, MN, 2009, http://www.cs.umn.edu/~metis.
[32] B.S. Kirk, J.W. Peterson, R.H. Stogner, G.F. Carey, libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations, Eng. Comput. 22 (2006) 237–254.
[33] W. Klieber, B. Rivière, Adaptive simulations of two-phase flow by discontinuous Galerkin methods, Comput. Methods Appl. Mech. Eng. 196 (2006) 404–419.
[34] D.A. Knoll, D.E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[35] J. Kou, S. Sun, Convergence of discontinuous Galerkin methods for incompressible two-phase flow in heterogeneous media, SIAM J. Numer. Anal. 51 (2013) 3280–3306.
[36] J. Kou, S. Sun, Upwind discontinuous Galerkin methods with conservation of mass of both phases for incompressible two-phase flow in porous media, Numer. Methods Partial Differ. Equ. 30 (2014) 1674–1699.
[37] K.A. Lie, An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST), Cambridge University Press, Cambridge, 2019.
[38] H. Liu, K. Wang, Z. Chen, A family of constrained pressure residual preconditioners for parallel reservoir simulations, Numer. Linear Algebra Appl. 23 (2016) 120–146.
[39] L. Liu, D.E. Keyes, S. Sun, Fully implicit two-phase reservoir simulation with the additive Schwarz preconditioned inexact Newton method, in: SPE Reserv. Charact. Simul. Conf. Exhib., Society of Petroleum Engineers, 2013.
[40] L. Luo, W.S. Shiu, R. Chen, X.-C. Cai, A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis, J. Comput. Phys. 399 (2019) 108926.
[41] A. Michel, A finite volume scheme for the simulation of two-phase incompressible flow in porous media, SIAM J. Numer. Anal. 41 (2003) 1301–1317.
[42] J.E.P. Monteagudo, A. Firoozabadi, Comparison of fully implicit and IMPES formulations for simulation of water injection in fractured and unfractured media, Int. J. Numer. Methods Eng. 69 (2007) 698–728.
[43] R.D. Nair, H.W. Choi, H.M. Tufo, Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core, Comput. Fluids 30 (2009) 309–319.
[44] D. Nayagum, G. Schäfer, R. Mosé, Modelling two-phase incompressible flow in porous media using mixed hybrid and discontinuous finite elements, Comput. Geosci. 8 (2004) 49–73.
[45] D.W. Peaceman, Fundamentals of Numerical Reservoir Simulation, Elsevier, Amsterdam, 1977.
[46] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig, A mixed-dimensional finite volume method for two-phase flow in fractured porous media, Adv. Water Resour. 29 (2006) 1020–1036.
[47] B. Rivière, M.F. Wheeler, K. Banas, Part II. Discontinuous Galerkin method applied to a single phase flow in porous media, Comput. Geosci. 4 (2000) 337–349.
[48] B. Rivière, M.F. Wheeler, Miscible displacement in porous media, in: S.M. Hassanizadeh, R.J. Schotting (Eds.), Proceedings of the XIV International Conference on Computational Methods in Water Resources, Elsevier, Amsterdam, 2002.
[49] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (1993) 461–469.
[50] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, PA, 2003.
[51] J.N. Shadid, R.S. Tuminaro, K.D. Devine, G.L. Hennigan, P.T. Lin, Performance of fully coupled domain decomposition preconditioners for finite element transport/reaction simulations, J. Comput. Phys. 205 (2005) 24–47.
[52] B. Smith, P. Bjørstad, W. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, 1996.
[53] S. Sun, M.F. Wheeler, Discontinuous Galerkin methods for coupled flow and reactive transport problems, Appl. Numer. Math. 52 (2005) 273–298.
[54] S. Sun, M.F. Wheeler, Symmetric and nonsymmetric discontinuous Galerkin methods for reactive transport in porous media, SIAM J. Numer. Anal. 43 (2005) 195–219.
[55] A. Toselli, O. Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer, Berlin, 2005.
[56] K. Wang, H. Liu, Z. Chen, A scalable parallel black oil simulator on distributed memory parallel computers, J. Comput. Phys. 301 (2015) 19–34.
[57] C. Yang, X.-C. Cai, Parallel multilevel methods for implicit solution of shallow water equations with nonsmooth topography on cubed-sphere, J. Comput. Phys. 230 (2011) 2523–2539.
[58] C. Yang, X.-C. Cai, A scalable fully implicit compressible Euler solver for mesoscale nonhydrostatic simulation of atmospheric flows, SIAM J. Sci. Comput. 36 (2014) S23–S47.
[59] H. Yang, X.-C. Cai, Parallel fully implicit two-grid Lagrange-Newton-Krylov-Schwarz methods for distributed control of unsteady incompressible flows, Int. J. Numer. Methods Fluids 72 (2013) 1–21.
[60] H. Yang, C. Yang, S. Sun, Active-set reduced-space methods with nonlinear elimination for two-phase flow problems in porous media, SIAM J. Sci. Comput. 38 (2016) B593–B618.
[61] H. Yang, S. Sun, C. Yang, Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media, J. Comput. Phys. 332 (2017) 1–20.
[62] H. Yang, S. Sun, Y. Li, C. Yang, A scalable fully implicit framework for reservoir simulation on parallel computers, Comput. Methods Appl. Mech. Eng. 330 (2018) 334–350.
[63] H. Yang, S. Sun, Y. Li, C. Yang, A fully implicit constraint-preserving simulator for the black oil model of petroleum reservoirs, J. Comput. Phys. 396 (2019) 347–363.
[64] H. Yang, S. Sun, Y. Li, C. Yang, Parallel reservoir simulators for fully implicit complementarity formulation of multicomponent compressible flows, Comput. Phys. Commun. 244 (2019) 2–12.
[65] A. Zidane, A. Firoozabadi, An implicit numerical model for multicomponent compressible two-phase flow in porous media, Adv. Water Resour. 85 (2015) 64–78.