Accepted Manuscript

Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media

Haijian Yang, Shuyu Sun, Chao Yang

PII: S0021-9991(16)30628-3
DOI: <http://dx.doi.org/10.1016/j.jcp.2016.11.036>
Reference: YJCPH 6986

To appear in: *Journal of Computational Physics*

Received date: 30 March 2016
Revised date: 26 October 2016
Accepted date: 24 November 2016

Please cite this article in press as: H. Yang et al., Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media, *J. Comput. Phys.* (2016), <http://dx.doi.org/10.1016/j.jcp.2016.11.036>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- A variational inequality formulation of two-phase flow problem is proposed.
- The variational inequality ensures the physical feasibility saturation.
- A fully implicit method with adaptive dt is applied to achieve high stability.
- A nonlinear elimination preconditioner is used to accelerate nonlinear solve.
- Numerical results show the proposed approach is more robust than classical IMPES.

Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media

Haijian Yang^a, Shuyu Sun^b, Chao Yang^{c,d,*}

^aCollege of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, PR China

^bDivision of Mathematics and Computer Science and Engineering, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

^cInstitute of Software, Chinese Academy of Sciences, Beijing 100190, PR China

^dState Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, PR China

Abstract

Most existing methods for solving two-phase flow problems in porous media do not take the physically feasible saturation fractions between 0 and 1 into account, which often destroys the numerical accuracy and physical interpretability of the simulation. To calculate the solution without the loss of this basic requirement, we introduce a variational inequality formulation of the saturation equilibrium with a box inequality constraint, and use a conservative finite element method for the spatial discretization and a backward differentiation formula with adaptive time stepping for the temporal integration. The resulting variational inequality system at each time step is solved by using a semismooth Newton algorithm. To accelerate the Newton convergence and improve the robustness, we employ a family of adaptive nonlinear elimination methods as a nonlinear preconditioner. Some numerical results are presented to demonstrate the robustness and efficiency of the proposed algorithm. A comparison is also included to show the superiority of the proposed fully implicit approach over the classical Implicit Pressure-Explicit Saturation (IMPES) method in terms of the time step size and the total execution time measured on a parallel computer.

Keywords: Two-phase flow, Fully implicit method, Variational inequality, Semismooth Newton method, Nonlinear preconditioner, Parallel computing

1. Introduction

Flow in subsurface porous media impacts the daily life of human being directly. First of all, it affects the performance of petroleum reservoirs, one of the most important energy resources for us on earth; consequently reservoir simulation models are used extensively by oil and gas companies in the development of new petroleum fields as well as managing existing fields. For example, production forecasts are often needed for investment decisions in

*Corresponding author.

Email addresses: haijianyang@gmail.com (Haijian Yang), shuyu.sun@kaust.edu.sa (Shuyu Sun), yangchao@iscas.ac.cn (Chao Yang)

developed fields, which require computational simulations of multi-phase flow in petroleum reservoirs. In addition, subsurface flow and transport directly affects our environment, and there is a great need to provide accurate assessments of risk and engineering performance for key processes with far-reaching consequences. Two important examples are the computational studies on geological storage of nuclear waste and carbon dioxide; in both cases, simulation of subsurface flow and transport is used to predict the long-term behavior of natural or engineered materials aiming at performance and safety assessment. Because of the complexity of the porous media, the flowing fluid, and their interaction, it is crucial to develop accurate and efficient modeling and computational methodology of subsurface flow and transport for deeper understanding of the underlying physics and for optimized management of subsurface environmental resources.

In conventional modeling approaches for flow and transport in geological formation, Darcy's law is assumed for single-phase flow, and extended Darcy's law for multi-phase flow in the media. These two phenomenological laws together with conservation laws and fluid properties are often used to model the fluid flow behaviors in the subsurface system. Conservative finite difference methods (in particular the block-centered finite difference method) and conservative finite element methods (in particular the Raviart-Thomas mixed finite element method) are then used to discretize the Partial Differential Equations (PDE) system in space. To discretize the PDE system in time and to tackle the coupled equations, various decoupled splitting schemes and time integration schemes have been used, which are reviewed below.

Among the many solution schemes available, fully explicit methods (e.g. forward Euler methods) are easy to implement and computationally cheap for a single time step; however, due to the severe CFL condition imposed by the PDE with strong nonlinearity and stiffness, fully explicit methods are seldom used because enormously many time steps are required for a realistic simulation. One popular solution scheme people use in practice is the IMPES (implicit in pressure and explicit in saturation) scheme [9, 10, 11, 12, 13, 14, 23, 58]. This scheme takes advantage of the fact that the pressure information (thus the Darcy velocity as well) propagates much more rapidly through the domain than the information of fluid saturations. Often, a number of small saturation time steps are carried out immediately after a single large pressure time step, because saturation changes with time faster than pressure. Iterative IMPES [38, 42, 43] is an improved version of IMPES where a number of iterations are performed in a single pressure-saturation time step interval to increase its accuracy. Sequential method can be considered as another variation of IMPES, where all equations are solved in an implicit fashion without a full coupling of the system. It is believed that the most stable scheme for subsurface multi-phase flow is the fully implicit method [1, 15, 16, 44, 53], which is also known as the simultaneous solution method in reservoir simulation community, or simply the SS method. All the coupled nonlinear equations are solved simultaneously and implicitly in the SS method, usually using a Newton-type approach. Even though the SS method is unconditionally stable for large time steps under certain conditions, its computational cost and memory requirement might be expensive without a sophisticated nonlinear solver, especially when the size of problems becomes large. Adaptive implicit scheme [25] can be viewed as a variation of the SS method, where the expensive SS is confined

to those gridblocks that require it, while on the remaining gridblocks the IMPES scheme is implemented. Even with a decent adaptive implicit scheme, a sophisticated nonlinear solver is still crucial to achieve robust, accurate and efficient simulation, which is the subject of this paper. We also remark that this paper is restricted to the discussion of the two-phase flow, but the applied methodology is applicable to the other nonlinear multiphase flow problems in two or three dimensions.

A major challenge in the modeling and simulation of two-phase flow is the high nonlinearity of the mathematical model. Sources of the high nonlinearity in the model problem comes from heterogeneous permeability of high contrast, strong nonlinearities of relative permeability, and spatially varied capillary pressure functions. These nonlinearities are shared with more complex problems. Significantly, both in the model problem and for more realistic cases, the strong nonlinearities will mostly be confined to advancing fronts of limited spatial extent [51, 52]. As result, when solving such nonlinear systems, however, the predicted saturation can not frequently sit within physically meaningful range, i.e., undershoot or overshoot from the interval $[0, 1]$, such that the capillary pressure and relative permeability become undefined and thus leads to the failure of the process. A common approach to fix this issue is the application of a projection operator, to cut off the undershoot or overshoot of the solution from this interval. This treatment will destroy the mass conservation and thereby seriously ruin the numerical accuracy and physical interpretability of the simulation results [37]. In this work, for accurate modeling and simulation of two-phase flow, we take the inequality constraints into account and introduce a variational inequality (VI) formulation [21, 27] of two-phase flow by incorporating the corresponding PDE and the potential constraints, to naturally satisfy the basic boundedness requirement. Then, the two-phase variational inequality problem is formulated as a nonlinear complementarity problem [28, 48], which is solved by a family of semismooth Newton methods [17, 27, 46]. It is worth mentioning that most used algorithms in practice do not guarantee the computed saturation to stay within this range [10, 37, 38]. Very few efforts have been made in the design of numerical methods to insure positivity for two-phase flow problems; a recent example is [26].

The semismooth Newton method is popular for solving large sparse nonlinear systems of equations arising from discretization of variational inequality problem, and is quite robust and efficient. It is proved (see [47]) that, under certain assumptions, the semismooth Newton method enjoys the same properties as the classical smooth methods, such as locally super-linear convergence. When the nonlinearities in the system are well-balanced and a good initial guess is available, a near quadratic convergence is usually observed. However, if the nonlinearities of the system are strong and not well-balanced, such as the nonlinear system arising from the discretization of the two-phase flow in porous media, the convergence of the Newton method can be problematic. To overcome this difficulty, we develop a class of adaptive nonlinear elimination (NE) preconditioners to implicitly remove these components that cause troubles for the Newton method. The NE algorithm is applied in the intermediate Newton solution to identify these components to be eliminated before a new global Newton iteration, so that the overall performance of the Newton-based kernel solver is improved. The NE method in conjunction with the inexact Newton method [18, 19, 20] has

been applied successfully to some smoothly nonlinear systems (i.e., the nonlinear function is smooth), such as the shocked duct flow [39] and the transonic full potential equation [31, 32]. However, designing effective elimination strategies for variational inequality problems is not trivial, since the corresponding residual function is not differentiable everywhere. In this paper, we extend the applicability of NE for variational inequality solution of two-phase flow in porous media, and perform the intensive numerical experiments to obtain insightful observation on the design of elimination processes.

The remainder of the paper is organized as follows. A variational inequality based model of the two-phase flow problem is introduced in Section 2. In Section 3, we briefly review the semismooth Newton for solving general variational inequality problems. In Section 4, we describe in detail the proposed nonlinearly preconditioned semismooth Newton algorithm. Three sets of numerical experiments with parallel performance results are reported in Section 5 and some concluding remarks are given in Section 6.

2. Mathematical model and discretization

In the section, we present a mathematical model for the two-phase flow problem in a heterogeneous media under incompressible conditions. This is the prototypical system for a wetting phase ($\alpha = w$) and a non-wetting ($\alpha = n$), which is described by the saturation equation and Darcy's law of the wetting and non-wetting phases. The mass balance of each phase is given by a hyperbolic saturation equation,

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{v}_\alpha = q_\alpha, \quad \alpha = w, n, \quad (1)$$

where ϕ is the porosity of the porous media, S_α denotes the saturation, $\mathbf{v}_\alpha$ is the velocity, and q_α represents the source term. According to Darcy's law, the velocity for each phase is determined by

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K} (\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad \alpha = w, n, \quad (2)$$

where $\mathbf{K}$ is the absolute permeability tensor of the porous media, p_α is the pressure, ρ_α is the density, and $\mathbf{g} = g \nabla z$ with g the gravity acceleration constant and z the depth at the position x . The mobility function $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$ is the ratio of the relative permeability $k_{r\alpha}$ and the viscosity μ_α . The void of porous medium is jointly filled by the two fluids, and the relation of their saturations satisfies the following constraints

$$\begin{cases} S_w \geq 0, S_n \geq 0, \\ S_w + S_n = 1. \end{cases} \quad (3)$$

The relation between the wetting and non-wetting phase pressures is given by the capillary pressure [10, 30],

$$p_c(S_w) = p_n - p_w. \quad (4)$$

The total mobility is expressed as $\lambda_t = \lambda_w + \lambda_n$, and we also define $q_t = q_w + q_n$.

Then, the potential equation of the two-phase flow problem is reformulated as

$$-\nabla \cdot (\lambda_t \mathbf{K} \nabla p_w) - \nabla \cdot (\lambda_n \mathbf{K} \nabla p_c) + \nabla \cdot ((\lambda_n \rho_n + \lambda_w \rho_w) \mathbf{K} \mathbf{g}) = q_t. \quad (5)$$

And the wetting-phase saturation equation becomes

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) = q_w. \quad (6)$$

Suppose the boundary of the computational domain Ω is composed of two parts $\partial\Omega = \Gamma^D + \Gamma^N$ with $\Gamma^D \cap \Gamma^N = \emptyset$. The boundary conditions associated to (5) and (6) are

$$\begin{cases} p_w = p^D & \text{on } \Gamma^D, \\ \mathbf{v} \cdot \mathbf{n} = q^N & \text{on } \Gamma^N, \\ S_w = S^N & \text{on } \partial\Omega. \end{cases}$$

In (6), the wetting-phase saturation S_w need to satisfy the potential constraints (3). Here, we take the constraints into account and introduce a variational inequality problem for the two-phase flow. Considering a L^2 -inner product and the constraints for the saturation S_w , we obtain the following variational inequality, see the references [5, 21, 27, 29] for more details.

Problem 1 Given $S_w(\cdot, 0) = S_w^0 \in H^1(\Omega)$ with $0 \leq S_w^0 \leq 1$, find $S_w \in H^1(\Omega_T)$ such that $0 \leq S_w \leq 1$ a.e. in $\Omega_T = \Omega \times [0, T]$ and

$$\left\langle \phi \frac{\partial S_w}{\partial t}, \mathcal{X} - S_w \right\rangle + \left\langle \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})), \mathcal{X} - S_w \right\rangle - \left\langle q_w, \mathcal{X} - S_w \right\rangle \geq 0, \quad (7)$$

which has to hold for almost all t and all $\mathcal{X} \in H^1(\Omega)$ with $0 \leq \mathcal{X} \leq 1$. Here $\langle \cdot, \cdot \rangle$ denotes the L^2 -inner product.

In the following, with the help of Lagrange multipliers μ_- and μ_+ for the inequality constraints $S_w \geq 0$ and $S_w \leq 1$, respectively, we reformulate the inequality (7) as follows.

Theorem 1 A function $S_w \in L^2(0, T; H^2(\Omega)) \cap H^1(\Omega_T)$ solves the inequality (7) if there exists $\mu_-, \mu_+ \in L^2(\Omega_T)$ such that

$$\begin{cases} \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w + \mu_+ - \mu_- = 0, \\ 0 \leq S_w \leq 1, \\ \mu_+(S_w - 1) = 0, \quad \mu_-(S_w - 0) = 0, \\ \mu_+ \geq 0, \quad \mu_- \geq 0, \end{cases} \quad (8)$$

where all equalities and inequalities have to hold almost everywhere.

Proof. Let $\mathcal{Y} \in L^2(0, T; H^2(\Omega))$ and hold the constraints $0 \leq \mathcal{Y} \leq 1$ a.e. in Ω_T . Multiplying the first equation of (8) by the term $\mathcal{Y} - S_w$, we obtain

$$\int_{\Omega_T} \left(\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w \right) (\mathcal{Y} - S_w) + \int_{\Omega_T} \mu_+ (\mathcal{Y} - S_w) - \int_{\Omega_T} \mu_- (\mathcal{Y} - S_w) = 0.$$

Using the properties of $\mathcal{Y}$ and the last three equations of (8), we have

$$\mu_+ (\mathcal{Y} - S_w) \leq 0, \quad \mu_- (\mathcal{Y} - S_w) \geq 0, \quad \text{a.e. in } \Omega_T.$$

Then the following inequality holds

$$\int_{\Omega_T} \left(\phi \frac{\partial S_w}{\partial t} \right) (\mathcal{Y} - S_w) + \int_{\Omega_T} (\nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w) (\mathcal{Y} - S_w) \geq 0.$$

For any given $t \in [0, T]$, by setting

$$\mathcal{Y} = \begin{cases} \mathcal{X} & \text{if } t \in (t - \epsilon, t + \epsilon), \\ S_w & \text{else,} \end{cases}$$

we complete the localization in time and then the inequality (7) follows. $\square$

Corollary 1 *A function $S_w \in L^2(0, T; H^2(\Omega)) \cap H^1(\Omega_T)$ solves the inequality (7) if it is the solution of the following problem*

$$\begin{cases} S_w = 0, \quad \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w \geq 0, \\ S_w = 1, \quad \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w \leq 0, \\ S_w \in (0, 1), \quad \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_w \mathbf{K} (\nabla p_w - \rho_w \mathbf{g})) - q_w = 0, \end{cases} \quad (9)$$

where all equalities and inequalities have to hold almost everywhere.

Proof. It follows from the fact that the problem (8) is equivalent to the system (9) by eliminating Lagrange multipliers μ_- and μ_+ . $\square$

In this paper, we employ a mixed finite element method (MFEM) for the spatial discretization. Let Ω be the computational domain covered with a finite number of mesh cells. We use the Raviart-Thomas space RT_κ for the velocity unknown in this paper. The corresponding space for the pressure unknown is Q_κ , the space of piecewise tensor-product polynomials of order κ . The Raviart-Thomas space RT_κ of order κ is the smallest polynomial space such that the divergence maps RT_κ onto Q_κ . Even though the work can be extended to higher-order $\kappa \geq 1$, we consider only the lowest-order approximation (i.e. $\kappa = 0$) in the

numerical examples of this paper. We denote the finite-dimensional velocity space (which vanishes on Γ_N) by $\mathbf{V}_h$; and denote the finite-dimensional pressure space by W_h . We first define the $\mathbf{U}_a$ operator acting on (p_w, S_w) by

$$\begin{aligned} \mathbf{U}_a(p_w, S_w) &\in \mathbf{V}_h + E(q^N), \\ (\lambda_t^{-1}(S_w) \mathbf{K}^{-1} \mathbf{U}_a(p_w, S_w), \mathbf{v}) &= -\langle p^D, \mathbf{v} \cdot \mathbf{n} \rangle_{\Gamma^D} + (p_w, \nabla \cdot \mathbf{v}), \\ &\quad \forall \mathbf{v} \in \mathbf{V}_h, \end{aligned}$$

where $E(q^N)$ is an extension of the Neumann boundary data for $\mathbf{v} \cdot \mathbf{n}$ into the entire domain. We then define the $\mathbf{U}_c$ operator acting on (p_w, S_w) by

$$\begin{aligned} \mathbf{U}_c(p_w, S_w) &\in \mathbf{V}_h, \\ (\mathbf{K}^{-1} \mathbf{U}_c(p_w, S_w), \mathbf{v}) &= ((\rho_n \lambda_n(S_w) + \rho_w \lambda_w(S_w)) \mathbf{g}, \mathbf{v}) + \\ &\quad (p_c(S_w), \nabla \cdot (\mathbf{v} \lambda_n(S_w))), \quad \forall \mathbf{v} \in \mathbf{V}_h. \end{aligned}$$

Here, the $\mathbf{U}_a$ operator is used to define the total Darcy velocity excluding the influence of gravity and capillarity, while the $\mathbf{U}_c$ operator is used to define the total Darcy velocity contributed from gravity and capillarity.

For a given time-stepping sequence $0 = t^{(0)} < t^{(1)} < t^{(2)} < \dots < t^{(M)} = T_f$, we define the time step size $\Delta t^{(m)} = t^{(m+1)} - t^{(m)}$ and use the superscript (m) to denote the discretized evaluation at time point $t = t^{(m)}$. At each time step $(t^{(m)}, t^{(m+1)})$, the fully implicit mixed finite element reads: we seek $(p_w^{(m+1)}, S_w^{(m+1)}) \in W_h \times W_h$ such that

$$\begin{aligned} (F^{(p_w)}(p_w^{(m+1)}, S_w^{(m+1)}), w) &= 0, \quad \forall w \in W_h, \\ (F^{(S_w)}(p_w^{(m+1)}, S_w^{(m+1)}), w) &= 0, \quad \forall w \in W_h, \end{aligned}$$

where the functions $F^{(p_w)}$ and $F^{(S_w)}$ are defined as

$$\begin{aligned} &F^{(p_w)}(p_w^{(m+1)}, S_w^{(m+1)}) \\ &= \nabla \cdot \mathbf{U}_a(p_w^{(m+1)}, S_w^{(m+1)}) + \nabla \cdot \mathbf{U}_c(p_w^{(m+1)}, S_w^{(m+1)}), \\ &\quad (F^{(S_w)}(p_w^{(m+1)}, S_w^{(m+1)}), w) \\ &= (q_w, w) - \left(\phi \frac{S_w^{(m+1)} - S_w^{(m)}}{\Delta t^{(m)}}, w \right) \\ &\quad - \sum_E \left\langle \frac{\lambda_w(S_w^{(m+1),*})}{\lambda_t(S_w^{(m+1),*})} \mathbf{U}_a(p_w^{(m+1)}, S_w^{(m+1)}) \cdot \mathbf{n}_{\partial E} + \lambda_w(S_w^{(m+1),*}) \rho_w^* \mathbf{K} \mathbf{g} \cdot \mathbf{n}_{\partial E}, w \right\rangle_{\partial E}. \end{aligned}$$

Here $S_w^{(m+1),*}$ is the upwind value of $S_w^{(m+1)}$ according to $\mathbf{U}_a(p_w^{(m+1)}, S_w^{(m+1)})$. That is, since $S_w^{(m+1)}$ is a element-wise constant function, $S_w^{(m+1)}$ is not uniquely defined on element interfaces, where it can take the value from either of the two neighboring elements. We use

the velocity $\mathbf{U}_a(p_w^{(m+1)}, S_w^{(m+1)})$ to categorize the two neighboring elements as one upstream element and one downstream element. The upwind value of $S_w^{(m+1)}$ is its value from the upstream element. Similarly, ρ_w^* is the upwind value of ρ_w according to the wetting-phase velocity

$$\frac{\lambda_w(S_w^{(m+1),*})}{\lambda_t(S_w^{(m+1),*})} \mathbf{U}_a(p_w^{(m+1)}, S_w^{(m+1)}) + \lambda_w(S_w^{(m+1),*}) \rho_w \mathbf{K} \mathbf{g}.$$

Now we denote $X = (p_w, S_w)^T$, the above fully implicit formulation results in a nonlinear algebraic system:

$$F(X) = \begin{pmatrix} F^{(p_w)}(X) \\ F^{(S_w)}(X) \end{pmatrix} = 0, \quad (10)$$

for each time step. The primary variables in the system (10) are p_w and S_w , for simplicity in the following the subscript w is ignored. Moreover, the discretization of variational inequality (9) is given as

$$\begin{cases} S = 0 & \text{and } F^{(S)}(X) \geq 0, \\ S = 1 & \text{and } F^{(S)}(X) \leq 0, \\ S \in (0, 1) & \text{and } F^{(S)}(X) = 0, \end{cases} \quad (11)$$

with only one of these three equations holding at a time. The system (11) is known as a box constrained variational inequality or mixed complementarity problem, see [21, 27, 28, 46, 48] and references therein.

For the long-term simulation of two-phase flow in porous media, the evolution of the system usually admits various time scales and the calculation often lasts for a long time. Therefore, it is necessary to use an adaptive time step control for the numerical simulation. In the study, analogous to the switched evolution/relaxation strategy in [45], we start with a relatively small time step size $\Delta t^{(0)}$, and update its value via $\Delta t^{(m+1)} = \beta^{(m)} \Delta t^{(m)}$ with $\beta^{(m)}$ being the scaling factor of the adjacent time step size

$$\beta^{(m)} = \max \left(\frac{1}{\theta_1}, \min \left(\theta_1, \left(\frac{\|F(X^{(m)})\|}{\|F(X^{(m+1)})\|} \right)^{\theta_2} \right) \right), \quad (12)$$

where $\|F(X^{(m+1)})\|$ is the Euclidean norm of $F(X^{(m+1)})$. In the formulation, $\theta_1 \in (0, +\infty)$ is a safeguard to avoid excessive change of the time step size between any two immediate time steps, and $\theta_2 \in (0, 1]$ is used to control the adjustment of the time step size. In addition, we use the fully coupled ordering to build up the large sparse nonlinear algebraic system of equations, by which we mean that all two variables defined at the same mesh point are always together throughout the calculations. At each mesh point, we arrange the unknowns in the order of p_{ij} and S_{ij} , and then all mesh points numbered in the natural ordering. That is, the unknowns are ordered in the order of

$$X = (p_{11}, S_{11}, \dots, p_{ij}, S_{ij}, \dots, p_{n_x n_y}, S_{n_x n_y}) = (X_1, \dots, X_N) \in \mathbb{R}^N$$

and the corresponding functions are ordered in the order of

$$F(X) = (F_{11}^{(p)}, F_{11}^{(S)}, \dots, F_{ij}^{(p)}, F_{ij}^{(S)}, \dots, F_{n_x n_y}^{(p)}, F_{n_x n_y}^{(S)}) = F(X_1, \dots, X_N) \in \mathbb{R}^N,$$

where $N = 2n_x n_y$. Moreover, we define a lower bound vector by

$$\phi = (-\infty, 0, \dots, -\infty, 0, \dots, -\infty, 0) \in \mathbb{R}^N,$$

and a upper bound vector

$$\psi = (+\infty, 1, \dots, +\infty, 1, \dots, +\infty, 1) \in \mathbb{R}^N.$$

Let $\mathcal{I} = \{1, 2, \dots, N\}$ be an index set with each index corresponding to an unknown component X_i and a nonlinear residual component F_i . Then the variational inequality of the nonlinear system (10) is defined as follows: find $X \in \mathbb{R}^N$ such that one of the following holds for each $i \in \mathcal{I}$,

$$\begin{cases} X_i = \phi_i & \text{and } F_i(X) \geq 0, \\ X_i = \psi_i & \text{and } F_i(X) \leq 0, \\ X_i \in (\phi_i, \psi_i) & \text{and } F_i(X) = 0, \end{cases} \quad (13)$$

where the inequality holds componentwisely by $\phi = (\phi_1, \dots, \phi_n)^T$ and $\psi = (\psi_1, \dots, \psi_n)^T$, i.e., only one of these three equations hold at a time. A special case of (13) is the nonlinear complementarity problem (NCP) by taking $\psi_i = +\infty$ for each i ,

$$\begin{cases} \text{find} & X \in \mathbb{R}^N, \\ \text{such that} & X \geq \phi, \quad F(X) \geq 0, \quad (X - \phi)^T F(X) = 0. \end{cases} \quad (14)$$

Moreover, if the lower bound vector $\phi = -\infty$, then the complementarity problem is reduced to the classical nonlinear system (10).

3. The classical semismooth Newton algorithms

We briefly review the framework of the semismooth Newton algorithm for solving variational inequality problems [17, 27, 46, 48, 56], which serves as the basis of the proposed method. The semismooth algorithm is based on a reformulation of (13) as a semismooth system of equations satisfying a semismooth property using the NCP-function [28]. In the following, we first recall the reconstruction of (14) by using the Fischer-Burmeister function [24], and then focus on the its application in the variational inequality problem (13).

A function $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}$ is called an NCP-function if it satisfies the following property

$$\varphi(a, b) = 0 \iff a \geq 0, b \geq 0, ab = 0.$$

Then, for any NCP function φ , the complementarity problem (14) is reformulated as the following nonlinear system of equations

$$\mathcal{H}(X) = \begin{pmatrix} \varphi(X_1 - \phi_1, F_1(X)) \\ \vdots \\ \varphi(X_i - \phi_i, F_i(X)) \\ \vdots \\ \varphi(X_N - \phi_N, F_N(X)) \end{pmatrix} = 0, \quad (15)$$

where $\mathcal{H} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is componentwise. In the study, we use the Fischer-Burmeister function to build the nonlinear system (15),

$$\varphi(a, b) := a + b - \sqrt{a^2 + b^2}.$$

Semismooth Newton methods, such as the ones studied in [17, 33, 34, 55], can be seen as the application of a Newton-type method with a generalized Jacobian to the system (15). In particular, for the Fischer-Burmeister function, the generalized Jacobian matrix is of the form

$$D_a(X) + D_b(X)\nabla F(X),$$

where nonnegative diagonal matrices

$$D_a = \text{diag}(d_{a_1}, \dots, d_{a_n}), \quad D_b = \text{diag}(d_{b_1}, \dots, d_{b_n}) \quad (16)$$

consist of the partial derivatives of the mapping φ with respect to the first variable $a = X - \phi$ and the second variable $b = F(X)$, respectively, or a suitable approximation to these partial derivatives at those points where φ is not differentiable [22]. Hence, let $a_i = X_i - \phi_i$ and $b_i = F_i(X)$, the values of D_a and D_b in (16) corresponding to the Fischer-Burmeister function take the following form.

- If $a_i^2 + b_i^2 \neq 0$, then

$$\begin{cases} d_{a_i} = 1 - \frac{a_i}{\sqrt{a_i^2 + b_i^2}}, \\ d_{b_i} = 1 - \frac{b_i}{\sqrt{a_i^2 + b_i^2}}. \end{cases}$$

- If $a_i^2 + b_i^2 = 0$, then

$$(d_{a_i}, d_{b_i}) \in \left\{ (1 - \alpha_1, 1 - \alpha_2) \mid \sqrt{\alpha_1^2 + \alpha_2^2} \leq 1 \right\},$$

for any $\alpha_1 \geq 0, \alpha_2 \geq 0$.

Similar to the complementarity problem (14), we reformulate the variational inequality problem (13) as the following nonlinear system [4, 46],

$$\mathcal{F}(X) = \begin{pmatrix} \varphi(X_1 - \phi_1, -\varphi(\psi_1 - X_1, -F_1(X))) \\ \vdots \\ \varphi(X_i - \phi_i, -\varphi(\psi_i - X_i, -F_i(X))) \\ \vdots \\ \varphi(X_N - \phi_N, -\varphi(\psi_N - X_N, -F_N(X))) \end{pmatrix} = 0,$$

where $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is componentwise. And we can define the corresponding generalized Jacobian matrices with respect to the function $\mathcal{F}(X)$ in a similar way. A major advantage of this reformulation is that the merit function

$$\Psi(X) := \frac{1}{2} \mathcal{F}(X)^T \mathcal{F}(X)$$

is continuously differentiable although the equation operator itself is nonsmooth [34].

The class of semismooth Newton methods for solving (13) is an iterative process. Given an initial guess $X^0 \in \mathbb{R}^N$ and let X^k be the current approximation at the k^{th} Newton iteration. Then the class of semismooth Newton with backtracking (SNB) algorithms [3, 46, 56] consists of the following steps to find the next approximation X^{k+1} .

- Find the inexact Newton direction s^k by solving the Jacobian system

$$\mathcal{F}(X^k) + J_k s^k = 0, \quad (17)$$

where J_k is a generalized Jacobian matrix.

- Compute $\bar{l}$ by a line search algorithm [18, 19] as the smallest l in $\{0, 1, 2, \dots\}$ such that

$$\Psi(X^k + \gamma^{\bar{l}} s^k) \leq \Psi(X^k) + \sigma \gamma^{\bar{l}} (\nabla \Psi(X^k))^T s^k, \quad (18)$$

and set $\lambda^k = \gamma^{\bar{l}}$.

- Compute the new approximate solution $X^k = X^k + \lambda^k s^k$, and set $k = k + 1$.

We remark that there are two major differences between the semismooth Newton method and the inexact Newton method with backtracking (INB) [18, 19, 20]. One is that the nonlinear function is modified in the semismooth Newton method. And the other is that the generalized Jacobian matrix is used instead of the classical one in the semismooth Newton method. Hence, the semismooth Newton method degrades to the INB for solving (13) with $\phi = -\infty$ and $\psi = +\infty$.

4. Nonlinearly preconditioned semismooth Newton algorithms

The convergent behavior of the semismooth Newton method is problem dependent. In this paper, we propose a class of nonlinearly preconditioned semismooth Newton algorithms to solve the multi-component nonlinear system (3), i.e., a nonlinear elimination (NE) method is proposed to precondition the outer semismooth Newton iterations, which includes a subspace correction step and a global update step.

Let the notation $X^{k+1} = \mathbf{M}(\mathcal{F}, X^k)$ be the use of a nonlinear stationary solver $\mathbf{M}$ to obtain an approximate solution, where $\mathcal{F}(X) = 0$ is a nonlinear system and X^k is the numerical solution at the k -th step. In general, the nonlinear elimination preconditioned semismooth Newton algorithm with backtracking (SNB-NE) can be described as follows: Let X^0 be an initial guess and assume that X^k is the current approximation, then the approximate solution at next iteration X^{k+1} can be obtained in a multiplicative fashion as

$$X^{k+1} = \mathbf{M}(\mathcal{F}, \mathbf{N}(\mathcal{G}, X^k)),$$

where $\mathcal{G}(X)$ is a nonlinear function for the nonlinear preconditioning, $Y = \mathbf{N}(\mathcal{G}, X)$ represents the application of the NE preconditioner when X is given, and then $X = \mathbf{M}(\mathcal{F}, Y)$ is

the use of the SNB method provided that Y is available. Hence, the proposed method can be seen as a class of nonlinear right-preconditioning methods, see the references [6, 41] for more details. Below, we present a high level description of the basic algorithm for a general problem in **Algorithm 1**.

Algorithm 1 Nonlinear elimination preconditioned semismooth Newton algorithm with backtracking (SNB-NE)

STEP 0. Given an initial guess $X^0 \in \mathbb{R}^N$ and set $k = 0$.

Until convergence do

STEP 1. (Subspace correction by NE):

- Evaluate $Y^k = \mathbf{N}(\mathcal{G}, X^k)$.
- Update $X^k = Y^k$.

STEP 2. (Global update by SNB):

- Inexactly solve $\mathcal{F}(X^k) + J_k s^k = 0$.
- Update $X^{k+1} = X^k + \lambda^k s^k$, where $\lambda^k \in (0, 1]$.

Set $k = k + 1$.

End

4.1. Nonlinear elimination preconditioner

The proposed NE method is based on a field partition of the nonlinear system $\mathcal{G}(X)$, which is used to split the equation conformally into two nonoverlapping components by physical or algorithmic aspects,

$$\mathcal{G}(X) = \mathcal{G}(u, v) = \begin{pmatrix} G(u, v) \\ H(u, v) \end{pmatrix} = 0, \quad (19)$$

where $G(u, v)$ and $H(u, v)$ are the nonlinear functions with respect to the variables u and v , respectively. Here, $H(u, v)$ denotes the nonlinear equations that have unbalanced nonlinearities. Then, based on the field partition, we define a new subspace function for the variables (u, v) ,

$$\tilde{F}(X) = \tilde{F}(u, v) = \begin{pmatrix} u - u_{old} \\ H(u, v) \end{pmatrix} = \begin{pmatrix} u - u_{old} \\ f(v) \end{pmatrix} = 0,$$

where u_{old} is the values of the previous iteration, i.e., the variable u keeps unchanged while the variable v is updated based on the system $f(v)$. Hence, in the above system, the nonlinear system $f(v) = 0$, as a submodel of $\mathcal{G}(X)$ with respect to the component v , requires to be solved in the following.

We first introduce some notations. Let $\mathcal{S}$ be an index set with respect to the component v , where each index corresponds to an unknown component v_i and a nonlinear function $f_i(v)$, i.e., the number of elements of $\mathcal{S}$ is the same as the dimension of the vector v , denoted by n_1 . Hence, the index set $\mathcal{S}$ is a subset of the index set $\mathcal{I} = \{1, 2, \dots, N\}$. For notational convenience, in the following we denote the set $\mathcal{S} = \{1, 2, \dots, n_1\}$. Then the index set is decomposed into two parts $\mathcal{S}_b$ (“b” for bad) with n_b components and $\mathcal{S}_g$ (“g” for good) with $n_1 - n_b$ components, i.e., $\mathcal{S} = \mathcal{S}_b \cup \mathcal{S}_g$. For this partition, we respectively define two subspaces,

$$V_b = \{v | v = (v_1, \dots, v_i, \dots, v_{n_1})^T \in \mathbb{R}^{n_1}, v_i = 0, \text{ if } i \notin \mathcal{S}_b\},$$

and

$$V_g = \{v | v = (v_1, \dots, v_i, \dots, v_{n_1})^T \in \mathbb{R}^{n_1}, v_i = 0, \text{ if } i \notin \mathcal{S}_g\}.$$

Then we can define the corresponding operators R_b and R_g , which restrict the vector v from $\mathbb{R}^{n_1}$ to V_b and V_g , respectively. Using the restriction operator R_b , we define the sub-nonlinear function $f_{\mathcal{S}_b} : \mathbb{R}^{n_1} \rightarrow V_b$ as $f_{\mathcal{S}_b}(v) = R_b(f(v))$, and define $T_b(v) : \mathbb{R}^{n_1} \rightarrow V_b$ as the solution of the following subspace nonlinear system

$$f_{\mathcal{S}_b}(R_g(v) + T_b(v)) = 0. \quad (20)$$

Then we may introduce a new global function

$$v_{new} = \mathcal{T}(v) = R_g(v) + T_b(v), \quad (21)$$

so that the good part of the vector v is kept while the bad part replaced by solving (20). Hence, the operation is referred to the so-called nonlinear elimination since some bad components are eliminated.

The key idea of NE is to remove the unbalanced nonlinearity before performing a global nonlinear update, so the effective identification of the bad components to be eliminated plays an important role in the success of the algorithm. Some physics-based or field-based strategy can be used to determine the subset indices $\mathcal{S}_b$ and $\mathcal{S}_g$. In the study, we use an adaptive nonlinear elimination method to smooth out these bad components so that the performance of the global nonlinear iteration can be improved. Let $\|f(v)\|_{\max}$ be the maximum norm of the residual function $f(v)$ and the component-wise residuals $|f_i(v)|$ with $i \in \mathcal{S}$, we use an algebraic-based approach to define the bad region of the component, which is determined by

$$|f_i(v)| > \frac{\rho}{N} \|f(v)\|_{\max}, \quad (22)$$

where $\rho > 0$ is a pre-chosen constant, N is the number of unknowns.

The role of ρ in (22) is used to control the number of “good” and “bad” components. When the index set $\mathcal{S}_g$ is empty, which is controlled by criterion (22) with $\rho = 0$, it means that all the equations with respect to the component v are bad in a global way. In the case, the proposed NE method is reduced to the classical field-split approach [41]. The effect of different values of ρ on the overall performance of SNB-NE will be investigated later in the numerical experiments.

Remark 1 For the simulation of two-phase flow in porous media using the nonlinear elimination method, we set the nonlinear function $\mathcal{G}(X)$ to the function $F(X)$ in (10), and split the multi-components nonlinear systems into two submodels with respect to the pressure and the saturation, respectively. Moreover, the saturation component is labelled as the to-be-eliminated component, i.e., $(u, v) = (p, S)$. In the process of the SNB-NE method, we first take several (controlled by a threshold N_{switch} , unless the line search fails) global nonlinear iterations using the SNB method, then solve the subspace problem by the inexact Newton method with backtracking, and this process continues until convergence.

4.2. The linear solver

The linear system problem,

$$AX = b,$$

has to be solved with different Jacobian matrices A and right hand side vectors b as a component of the proposed nonlinear solver, for both the linear Jacobian system and the nonlinear elimination process. Effective mechanisms for solving this system using iterative methods are indispensable and have a great impact upon the success of the algorithm. One of the keys to achieving good performance from an iterative method is to select an appropriate linear preconditioner. In this study, we employ an overlapping restricted additive Schwarz right-preconditioned Generalized Minimal RESidual (GMRES) method [7, 35, 49] for solving these systems.

To define the additive Schwarz preconditioner, we first partition the computational domain Ω into N_p nonoverlapping subdomains Ω_l (i.e., $\Omega_l \cap \Omega_{l'} = \emptyset, \forall l \neq l'$) for $l = 1, 2, \dots, N_p$, such that

$$\overline{\Omega} = \bigcup_{l=1}^{N_p} \overline{\Omega}_l,$$

where N_p is the number of subdomains and also the number of processor cores. Then within Ω we extend each subdomain Ω_l with δ layers of mesh cells to a larger subdomain Ω_l^δ that overlaps with its neighbors. In each overlapping subdomain, we define a local subdomain matrix A_l that is the restriction of the global matrix A to Ω_l^δ with the restriction operator R_l^δ , i.e.,

$$A_l = R_l^\delta A (R_l^\delta)^T.$$

Here, R_l^δ and $(R_l^\delta)^T$ are the restriction and prolongation operators, respectively. Given a global vector defined on all mesh cells in Ω , R_l^δ restricts the global vector to a local vector that is defined only on the mesh cells within the overlapping subdomain Ω_l^δ , while $(R_l^\delta)^T$ prolongates the restricted vector back to a vector defined on all mesh cells in the whole domain Ω with zeros filled to the unknowns corresponding to mesh cells outside Ω_l^δ . The one-level restricted additive Schwarz (RAS) preconditioner [8, 50, 54, 57] is defined as

$$M^{-1} = \sum_{l=1}^{N_p} (R_l^0)^T (A_l)^{-1} R_l^\delta, \quad (23)$$

where the restriction operator R_l^0 is defined as the restriction to put zeros at unknowns corresponding to mesh cells outside the non-overlapping subdomain Ω_l and the matrix-vector multiplication with $(A_l)^{-1}$ is calculated based on the sparse LU factorization of A_l .

5. Numerical experiments

We implement the algorithms studied in this paper using the open-source Portable, Extensible Toolkit for Scientific computation (PETSc) library ([2]). The numerical tests are carried out on the Tianhe-2 supercomputer. The computing nodes of Tianhe-2 are interconnected via a proprietary high performance network, and there are two 12-core Intel Ivy Bridge Xeon CPUs and 24GB local memory in each node. The Intel Xeon Phi coprocessors are not utilized in our tests. In the numerical experiments, we use all 24 CPU cores in each node and assign one subdomain to each core.

There are several nonlinear and linear iterative procedures in the proposed algorithms, and each requires a proper stopping condition. In the SNB-NE method, an absolute (relative) tolerance of 10^{-8} (10^{-6}) is utilized for the global nonlinear iteration, and the absolute (relative) tolerance is set to 10^{-10} (10^{-6}) for the subspace nonlinear iteration. In nonlinear iterations, we use the standard cubic backtracking algorithm [18, 19] with $\sigma = 10^{-4}$ in (18) to pick the step length. All linear systems are solved by the Schwarz preconditioned GMRES method with absolute and relative tolerances of 10^{-8} and 10^{-2} , respectively. In practice, we set the adaptivity parameters in (12) to be $\theta_1 = 2.5$ and $\theta_2 = 0.75$. Throughout this section, in the tables, “ N_p ” stands for the number of processors, which is the same as the number of subdomains, “Newton” is the average number of inexact Newton iterations per time step, “GMRES” is the average number of the preconditioned GMRES iterations per Newton iteration, and “Time” is the total compute time in seconds.

5.1. Test cases

In this paper, we conduct the numerical experiments based on four examples [30, 36, 52]. In these examples, we consider two-phase fluid flow in a horizontal layer with heterogeneous permeabilities. With the media being horizontal, the effect of gravity is neglected in all examples, except the last example. The void of medium is initially fully saturated with oil and then we flood the system by water at the left end. The production end is the right-hand side. The other boundaries are impermeable; that is, the normal component of the Darcy velocity on these boundaries vanishes. There is no injection/extraction to the interior of the domain. We use the absolute permeability tensor as $\mathbf{K} = kI$, where I is the identity matrix and k is a positive real number. In the tests, the capillary pressure function in (4) is given by

$$p_c = \frac{B_c}{\sqrt{k}} \log(S_e),$$

where B_c is a positive parameter and S_e is the normalized saturation that is chosen to be zero for the residual saturations of water and oil, i.e., $S_e = S_w$. The employed relative permeabilities are given by $k_{rw} = S_e^\beta$ and $k_{rn} = (1 - S_e)^\beta$ respectively, with β being a

positive integer number. In the computation, we assume the minimal value of the saturation is $S_w = 10^{-4}$, which is used to replace the lower bound 0 in (11).

In the first test case, as shown in the upper left panel of Figure 1, the tested medium with the domain dimensions $300\text{ m} \times 150\text{ m} \times 1\text{ m}$ consists of two subdomains with different configurations for the distribution of permeability,

$$k = \begin{cases} 100\text{ md} & 70 \leq x \leq 150 \text{ and } 50 \leq y \leq 100, \\ 1\text{ md} & \text{otherwise.} \end{cases}$$

The porosity of this medium is $\phi = 0.2$. We use the quadratic relative permeabilities, i.e., $\beta = 2$. The viscosities of water and oil are $\mu_w = 1\text{ cP}$ and $\mu_n = 0.45\text{ cP}$, respectively. The capillary pressure parameter B_c is taken as 70 bar. The injection rate is 0.15 PV/year.

In the second test case, the tested medium consists of three subdomains with the different permeabilities of 1 md or 100 md, i.e.,

$$k = \begin{cases} 1\text{ md} & 50 \leq y \leq 100, \\ 100\text{ md} & \text{otherwise.} \end{cases}$$

The configurations are shown in the upper right panel of Figure 1. The porosity of this medium ϕ is 0.2. The viscosities of water and oil are $\mu_w = 1\text{ cP}$ and $\mu_n = 0.5\text{ cP}$, respectively. The relative permeabilities are cubic, i.e., $\beta = 3$. The capillary pressure parameter B_c is taken as 60 bar. The injection rate is 0.2 PV/year.

The third and the fourth test cases both involve random permeabilities with $\phi = 0.2$, $\beta = 2$, $\mu_w = 1\text{ cP}$, $\mu_n = 0.2\text{ cP}$, and the injection rate of 0.1 PV/year. In the third test case, the dimension of tested medium is $100\text{ m} \times 100\text{ m} \times 1\text{ m}$ with the capillary pressure parameter of $B_c = 25$ and a random distribution of permeability in the range 0.03 to 400, as shown in the lower left panel of Figure 1. In the last test case, we consider a 3D tilted domain with a $300\text{ m} \times 150\text{ m} \times 150\text{ m}$ dimension and a random distribution of permeability with the range of $[0.0013, 161.6361]$, generated by a geostatistical model using the open source code MRST [40]. The densities of the wetting and non-wetting phases are $\rho_w = 1000\text{ kg/m}^3$ and $\rho_n = 600\text{ kg/m}^3$, respectively.

5.2. Numerical validation

We first validate the discretization scheme and the fully implicit solver by running Case-1 and Case-3. In Figure 2, we show the computed results obtained on a 100×50 mesh by using the SNB-NE and IMPES methods for Case-1. In this test, for the SNB-NE method we start with $\Delta t^{(0)} = 5 \times 10^{-3}$ and adjust the time step size by (12), and finish the simulation at $t = 3.334$ year after 98 time steps, leading to an average time step size of $\Delta t_{avg} = 0.03402$; for the IMPES method we use a fixed time step size $\Delta t = 5 \times 10^{-4}$ due to the stability constraints. We find that the numerical results of both the IMPES method and the fully implicit method are consistent to the published benchmark solutions in [36].

We then run Case-3 to verify the robustness of the fully implicit method. Figure 3 shows contour plots of wetting-phase saturation profiles by using SNB-NE at different times for

Figure 1: Heterogeneous permeabilities. We use the logarithmic scale for Case-3 and Case-4, i.e., $\log(k)$ where k has a unit of md.

Figure 2: Case-1: wetting-phase saturation profiles by using the SNB-NE and IMPES methods.

Case-3. In the second test, the mesh is 100×100 , the initial time step size to be $\Delta t^{(0)} = 10^{-2}$, and finish the simulation at $t = 3.573$ year after 53 time steps, i.e., the average time step size is $\Delta t_{avg} = 0.0674$. We can see that the fully implicit approach successfully resolves the evolution of the wetting-phase saturations. It is worth mentioning that, because of exceeding the physically feasible saturation fractions between 0 and 1, (a) for Case-1, the standard inexact Newton method with backtracking comes to a breakdown from the floating point exception when the time step size is larger than $\Delta t = 10^{-2}$ on a 100×50 mesh; and (b) for Case-3, the the inexact Newton method with backtracking fails when $\Delta t \geq 0.02$ on a 100×100 mesh.

Figure 3: Case-3: wetting-phase saturation profiles by using SNB-NE at different times.

In the simulation of two-phase flow in porous media, the flow behavior is influenced by the capillary pressure parameter B_c . In Figure 4, we show the wetting-phase saturation profiles with different capillary pressure parameters obtained on a 100×50 mesh and a initial time step size to be $\Delta t^{(0)} = 5 \times 10^{-3}$ for Case-2. The simulation is finished at 0.5 PVI, i.e., $t = 0.5/0.2$ year. It is observed that the flow behavior with homogeneous capillary pressure is essentially influenced by the low permeability field, which causes a great delay of displacement in the middle layer. Moreover, this delay of displacement becomes obvious with the decreasing of the capillary pressure. Table 1 shows the performance of the SNB-NE method with respect to the capillary pressure parameters. It is clearly seen that number of time steps and the execution time decrease as the decreasing of the capillary pressure.

Figure 4: Case-2: wetting-phase saturation profiles with different capillary pressures B_c .

Table 1: A comparison of different capillary pressure parameters for Case-2.

Capillary pressure	$B_c = 20$	$B_c = 40$	$B_c = 60$	$B_c = 80$
End of simulation time (years)	2.5226	2.5203	2.5082	2.5192
Number of time steps	82	75	72	71
Average time step size (years)	0.0308	0.0336	0.0348	0.0355
Average nonlinear iteration	9.8	9.4	9.3	9.6
Average linear iteration	7.7	7.5	7.4	7.3
Execution time (second)	42.0	39.2	38.0	37.5

Finally, we test the proposed method with Case-3, which is a 3D problem with random permeabilities as shown in the lower right panel of Figure 1. The heterogeneity of the permeabilities often causes unexpected early water breakthrough and is challenging for the numerical techniques. In the test we set the initial time step to $\Delta t = 10^{-3}$ and run the simulation on a $100 \times 50 \times 50$ mesh. The simulation is ended at $t = 1.5$ year. The contour plots of wetting-phase saturation profiles by using the SNB-NE method with different capillary pressures are shown in Figure 5. We can see that the proposed approach successfully resolves the rapid and abrupt evolution of the wetting-phase saturation, and the capillary pressure function has a significant influence on the simulation of the two-phase flow.

5.3. A comparison of SNB and SNB-NE

We present a comparison of the SNB-NE algorithm with the classical SNB method. As introduced in Section 2, the price to pay in using fully coupled implicit methods is to solve a nonlinear system at each time step. In order to maintain the physically feasible saturation fractions between 0 and 1, we propose a family of semismooth methods to ensure the saturation within physically meaningful range. However, since the nonlinear system is highly nonlinear, the SNB method often fails to converge due to the failure of linear search. Hence, in the study we use the SNB-NE method to fix this issue. Figures 6 and 7 show the histories of the nonlinear residuals of SNB and SNB-NE for the three test cases. We observe from the figures that the SNB-NE method converges well for the test cases. It is clear that the nonlinear preconditioner improves the performance of the global Newton iteration by balancing the nonlinearities.

In the following, we study the distribution of the bad components during the global nonlinear iterations. The effective identification of the bad components to be eliminated plays an important role in the success of the method. For many PDE problems with highly nonlinearity, the surface residual plot of the nonlinear functions for some variables provide such a piece of useful information to identify which field variable dominates the overall residual norm of the nonlinear system, which indicate that its nonlinearities are far from balanced. In Figure 8, we show the residual surface plots for the pressure and saturation components obtained by using the classical semismooth method at the first nonlinear iteration. One typical characteristic of these surface plots for Case-1 is that the residual error of the pressure component is quite small, and the overall residual norm is dominated by the saturation

Figure 5: Case-4: wetting-phase saturation profiles with different capillary pressures B_c . In the figure, “layer” denotes the layer of the mesh size in the z -direction.

Figure 6: Nonlinear residual history for Case-1 at the first (left) and second (right) time steps on a 100×50 mesh and with initial time step size $\Delta t^{(0)} = 5 \times 10^{-3}$.

Figure 7: Nonlinear residual history for Case-2 (left) and Case-3 (right) at the first time step. For Case-2, the initial time step size is $\Delta t^{(0)} = 5 \times 10^{-3}$ and the mesh size is 100×50 ; For Case-3, the initial time step size is $\Delta t^{(0)} = 10^{-2}$ and the mesh size is 100×100 .

component. As a result, many more Newton iterations are needed in order to try to remove this error. Hence, the saturation component S is recognized as the “bad component”. We remark that the residual surface plots for Case-2 and Case-3 are similar to that for Case-1, and we omit these plots for brevity.

Figure 8: Case-1: The residual surface plots for dominant components before first nonlinear iteration at the 40th time step. The mesh is 100×50 .

We further investigate how the “bad” region changes for the SNB–NE method during the time stepping. As shown in Figures 9 for Case-1, the distribution of the to-be-eliminated regions is marked as red, and the good region is marked as the gray. Note that the void of medium is initially fully saturation with oil and the we flood the system at the left end, and the production end is the right-hand side. Hence, the interface of the red color and the grey color in the figures can be seen as the mixing zone between water and oil. As the simulation moves forward, the water pushes the interface to the right-hand side correspondingly. It is worth mentioning that the performances of the to-be-eliminated regions for Case-2 and Case-3 are analogous to that for Case-1, and we also ignore these plots for brevity.

Now we study the effect of ρ in (22) for the SNB–NE method. The role of ρ is used to control the number of “good” and “bad” components. In Table 2, we show the effect of the parameter ρ for Case-1 and Case-2 with a 100×50 mesh. When the value of ρ is smaller, the region of the bad points will increase, meanwhile it leads to the dimension of the subspace nonlinear problem increases, in general the cost for solving subspace nonlinear

Figure 9: Case-1: Distribution of the to-be-eliminated regions (marked as red) by using SNB-NE at different times. The mesh is 100×50 . The initial time step size to be $\Delta t^{(0)} = 5 \times 10^{-3}$.

problems may decrease as well. A special case is $\rho = 0$, which means that all the points of the saturation component is bad in a global way. From Table 2, we find that $\rho \in [0, 10]$ is appropriate that compromises the costs between the global and subspace solves to minimize the overall compute time and guarantees the convergence of the totally method.

Table 2: The effect of ρ . The number of processors is $N_p = 4$. The mesh is 100×50 and the initial time step size to be $\Delta t^{(0)} = 5 \times 10^{-3}$. In the table, the columns of “SNB step” (“NE step”) denote the performance of the global (subspace) iteration, and “Time” in the the column of “SNB-NE” includes the total execution time for solving all the nonlinear systems for the global and subspace iterations.

Case	ρ	Newton	GMRES	Time	Newton	GMRES	Time	Time
		SNB step			NE step			SNB-NE
Case-1	0	9.9	13.9	40.2	4.22	1.79	12.2	52.4
	0.1	11.2	13.3	41.6	5.23	1.72	11.7	53.3
	1.0	10.1	13.4	38.1	4.48	1.75	10.2	48.3
	10	11.1	13.3	41.9	5.42	1.69	12.0	53.9
Case-2	0	10.7	8.0	26.7	9.55	1.73	17.7	44.4
	0.1	8.6	7.8	22.0	7.95	1.69	12.6	34.6
	1.0	9.4	7.4	25.1	7.88	1.61	12.0	37.1
	10	9.8	7.1	26.1	7.94	1.53	12.9	39.0

5.4. Parallel performance study

In this subsection, we mainly focus on the parallel performance of the SNB-NE method by using Case-2. An important feature of SNB-NE is the selections of linear preconditioners for the solution of global and subspace Jacobian systems. As introduced in Subsection 4.2, we consider the Schwarz preconditioned GMRES method for solving the global and subspace Jacobian systems. In the Schwarz preconditioner, the overlapping parameter δ plays an important role in controlling the number of linear iterations and the total compute time. In this test, we fix the overlapping size in the subspace Jacobian systems to one and investigate the effect of δ on the performance of the proposed method for the global iterations. For this experiment, we consider three meshes 1024×512 , 2048×1024 , and 4096×2048 , and use three corresponding initial time step size $\Delta t^{(0)} = 10^{-4}$, 10^{-5} , and 10^{-6} . The simulation is stopped after five implicit time steps, and the parameter ρ is fixed to zero. As shown in Table 3, from the numbers of linear iterations, we see that, in general, the algorithm converges better as the overlap increases. However, larger overlap also requires more compute time per iteration. We observe that the moderate overlap $\delta = 6$ provide a good trade off among the compute time and linear iterations. Note that the additive Schwarz preconditioner degenerates to a block-Jacobi preconditioner when the overlapping factor $\delta = 0$, and we find from Table 3 that the linear solver fails to converge when using the block-Jacobi preconditioner.

Table 3: Performance with respect to the overlapping factor δ for the global iterations. The “*” means the divergence of the method.

Mesh	N_p	δ	Newton	GMRES	Time
			SNB step		
1024×512	64	0	*	*	*
		2	13.8	21.4	25.3
		4	13.8	16.4	23.4
		6	13.8	13.6	22.4
		8	13.8	11.9	23.1
2048×1024	128	0	*	*	*
		2	8.2	43.5	61.7
		4	8.2	32.4	51.9
		6	8.2	26.7	48.0
		8	8.2	23.7	49.1
4096×2048	256	0	*	*	*
		2	5.4	94.9	212.8
		4	5.4	72.0	162.6
		6	5.4	60.4	140.8
		8	5.4	52.5	131.8

For high fidelity simulations on supercomputers with a large number of processors, the strong and weak scalabilities of an algorithm with respect to the number of processors are

critically important. The strong scalability is defined as how the execution time varies with the number of processors for a fixed total problem size, and the weak scalability is defined as how the execution time varies with the number of processors for a fixed problem size per processor. Here, we focus on the scalable performance of the SNB-NE method. We perform the strong scalability test on two fixed meshes and different number of processor cores in Table 4. In the test, the simulation is stopped after five implicit time steps for a 4096×2048 mesh and three implicit time steps for a 8192×4096 mesh with the same initial time step size $\Delta t^{(0)} = 10^{-6}$. We can see from the table that, and the number of linear iterations increases slowly with the growth of the the number of processor cores. In Figure 10, we report the compute time for the global and subspace iterations with respect to the number of processors. We see that the compute time of the proposed method decreases with the number of processors increases, and a good speedup is obtained from 64 to 4096 processors.

Table 4: Strong scalability with different numbers of processors N_p . In the table, the columns of “SNB step” (“NE step”) denote the performance of the global (subspace) iteration, and “Time” in the the column of “SNB-NE” includes the total execution time for solving all the nonlinear systems for the global and subspace iterations. The biggest simulation has $8192 \times 4096 \times 2 = 67,108,864$ degrees of freedom.

Mesh	N_p	Newton	GMRES	Time	Newton	GMRES	Time	Time
		SNB step			NE step			SNB-NE
4096×2048	64	5.4	42.0	546.1	2.8	1.0	36.6	582.7
	128	5.4	46.2	261.5	2.8	1.0	17.8	279.3
	256	5.4	60.4	140.8	2.8	1.0	8.8	149.6
	512	5.4	67.2	74.9	2.8	1.0	4.3	79.2
	1024	5.4	81.6	45.4	2.8	1.0	2.3	47.7
8192×4096	256	11.0	65.5	1169.8	3.7	1.0	35.8	1205.6
	512	10.7	74.2	530.4	3.7	1.0	16.6	547.0
	1024	10.7	103.2	308.5	3.7	1.0	8.4	316.9
	2048	11.0	123.2	167.3	3.7	1.0	4.3	171.6
	4096	11.0	144.9	100.2	3.7	1.0	2.5	102.7

A weak scalability test is carried out to examine the performance of the solver when the problem size is increased in proportion to the number of processor cores. Now let us compare the proposed fully implicit method with the IMPES method by solving Case-2 with strong nonlinear effect. In the test, we start with 8 processor cores and a 256×128 mesh, and the number of processor cores is increased as the mesh is refined accordingly. The simulation is terminated at 0.02 year. In the fully implicit method, the initial time step size $\Delta t^{(0)}$ is 2×10^{-3} for the case of a 256×128 mesh that leads to an average time step size of $\Delta t_{avg} = 5.29 \times 10^{-3}$, and then is reduced accordingly as the mesh is refined. In the semi-implicit method, the time step size is fixed to $\Delta t = 10^{-4}$ for the case of a 256×128 mesh, and then is reduced by half as the mesh is refined accordingly. However, the IMPES method is not convergent for the cases of a 1024×512 mesh with $\Delta t = 2.5 \times 10^{-4}$, because of floating point exception in the computation of the saturation variable. Hence, we choose

Figure 10: Strong scalability results with different number of processors N_p . The left panel corresponds to the SNB step, and the right panel corresponds to the NE step.

a smaller time step size for this case. As shown in Table 5, the new method is a clear winner when the mesh sizes become finer in terms of the time step size and the total execution time.

Table 5: Weak scaling results of the adaptive SNB-NE method and the IMPES method.

Method	Number of processors	8	32	128
	Mesh size	256×128	512×256	1024×512
SNB-NE	Initial time step size (years)	2×10^{-3}	6×10^{-4}	7×10^{-5}
	Average time step size (years)	5.29×10^{-3}	2.59×10^{-3}	3.71×10^{-4}
	Number of time steps	4	9	55
	Average nonlinear iteration	17.5	18.1	11.3
	Average linear iteration	2.9	6.9	13.5
	Execution time (second)	8.3	23.7	133.2
IMPES	Time step size (years)	10^{-4}	5×10^{-5}	1.25×10^{-5}
	Number of time steps	200	400	1600
	Execution time (second)	6.3	34.9	594.2

Finally, we analyze the behavior of the proposed method when the time step size is changed. In the test, we again run Case-2 on a fixed 512×256 mesh using 32 processor cores. The simulation is stopped at $t = 0.02$ year. The results on the average numbers of Newton and GMRES iterations as well as the total compute time for the global and subspace iterations are summarized in Table 6. The results in the table clearly indicate that the combination of nonlinear and linear iterations works well for even very large value of time steps.

6. Concluding remarks

In the paper, we proposed a new algorithm, namely a parallel nonlinearly preconditioned semismooth Newton method for variational inequality multi-components systems with high

Table 6: The effect of different time steps a 512×256 mesh. The number of processors is $N_p = 32$. The simulation is ended at $t = 0.02$. In the table, the columns of “SNB step” (“NE step”) denote the performance of the global (subspace) iteration, and “Time” in the column of “SNB-NE” includes the total execution time for solving all the nonlinear systems for the global and subspace iterations.

Δt_0	Steps	Δt_{avg}	Newton	GMRES	Time	Newton	GMRES	Time	Time
			SNB step			NE step			SNB-NE
10^{-3}	7	3.51×10^{-3}	25.1	5.1	17.4	23.6	1.3	9.9	27.3
8×10^{-4}	8	2.92×10^{-3}	20.7	6.4	16.8	17.3	1.3	7.2	24.0
6×10^{-4}	9	2.23×10^{-3}	18.1	6.9	17.0	14.2	1.3	6.7	23.7
2×10^{-4}	24	8.62×10^{-4}	10.9	8.2	29.1	7.7	1.2	9.5	38.6
8×10^{-5}	57	3.55×10^{-4}	6.3	10.0	45.2	4.2	1.0	12.2	57.4

nonlinearity, and applied it in two-phase flow simulations in a porous media. The key idea of the method is the use of a nonlinear elimination preconditioner to remove the the highly nonlinear components that often causes the failure of the nonlinear iteration for convergence. For the application of two-phase flow in porous media, we reformulate the two-phase model as a variational inequality that naturally ensures the physical feasibility of the saturation variable, and then solve it with the proposed SNB-NE method. We tested the algorithm for three benchmark problems and showed numerically that the fully implicit solver is more robust than the semi-implicit IMPES method and scalable in terms of parallelization on a supercomputer with thousands of processors.

Acknowledgment

The authors would like to express their appreciation to the anonymous reviewers for their invaluable comments, which have greatly improved the quality of the paper. The work was supported in part by Special Project on High-Performance Computing under the National Key R&D Program (2016YFB0200603) and National Natural Science Foundation of China (11571100, 91530323, 11272352). S. Sun was also supported by KAUST through the grant BAS/1/1351-01-01. C. Yang was also supported by Key Research Program of Frontier Sciences from CAS through the grant QYZDB-SSW- SYS006.

References

- [1] K. Aziz, A. Settari, Petroleum Reservoir Simulation, Applied Science Pub. London, 1979.
- [2] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Users Manual, Argonne National Laboratory, 2015.
- [3] S. J. Benson, T. S. Munson, Flexible complementarity solvers for large-scale applications, Optim. Methods Softw. 21 (2006) 155–168
- [4] S. C. Billups, Algorithms for complementarity problems and generalized equations. Ph.D.thesis, Computer Sciences Department, University of Wisconsin-Madison, Madison, WI, 1995.
- [5] L. Blank, H. Garcke, L. Sarbu, V. Styles, Primal-dual active set methods for Allen-Cahn variational inequalities with nonlocal constraints, Numer. Methods Partial Differ. Equ. 29 (2013) 999–1030.
- [6] P. Brune, M. Knepley, B. Smith, X. Tu, Composing scalable nonlinear algebraic solvers, SIAM Review, 57 (2015) 535–565.

- [7] 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.* 19 (1998) 246–265.
- [8] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM J. Sci. Comput.* 21 (1999) 792–797.
- [9] Z. Chen, G. Huan, B. Li, An improved IMPES method for two-phase flow in porous media, *Transport Porous Med.* 54 (2004) 361–376.
- [10] Z. Chen, G. Huan, Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, SIAM, Philadelphia, PA, 2006.
- [11] K. H. Coats, Reservoir simulation: State of art. *JPT*, 1633–1642, Aug. 1982, SPE 10200.
- [12] K. H. Coats, A note on IMPES and some IMPES-based simulation models, presented at the 15th symposium on reservoir simulation, Houston, TX, Feb. 1999, SPE 49774.
- [13] K. H. Coats, IMPES stability: the CFL limit, presented at the SPE Reservoir Simulation Symposium, Houston, TX. Mar. 2001, SPE 85956.
- [14] K. H. Coats, IMPES stability: selection of stable time steps, presented at the SPE Reservoir Simulation Symposium, Houston, TX. Feb. 2001, SPE 84924.
- [15] D. A. Collins, L. X. Nghiem, Y. K. Li, J. E. Grabenstetter, An efficient approach to adaptive implicit compositional simulation with an equation of state, *SPE Reservoir Engineering*, 7 (1992) 259–264.
- [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, *Computat. Geosci.* 1 (1997) 215–249.
- [17] T. De Luca, F. Facchinei, C. Kanzow, A semismooth equation approach to the solution of nonlinear complementarity problems, *Math. Prog.* 75 (1996) 407–439.
- [18] J. E. Dennis, R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.
- [19] S. C. Eisenstat, H. F. Walker, Globally convergent inexact Newton method, *SIAM J. Optim.* 4 (1994) 393–422.
- [20] S. C. Eisenstat, H. F. Walker, Choosing the forcing terms in an inexact Newton method, *SIAM J. Sci. Comput.* 17 (1996) 16–32.
- [21] I. Ekeland, R. Témam, *Convex Analysis and Variational Problems*, SIAM, Philadelphia, PA, 1999.
- [22] F. Facchinei, J. Soares, A new merit function for nonlinear complementarity problems and a related algorithm, *SIAM J. Opt.* 7 (1997) 225–247.
- [23] R. G. Fagin, C. H. Jr. Stewart, A new approach to the two-dimensional multiphase reservoir simulator, *SPE J.*, 1966, SPE 1188.
- [24] A. Fischer, A special Newton-type optimization method, *Optimization*, 24 (1992) 269–284.
- [25] P. A. Forsyth, Adaptive implicit criteria for two-phase flow with gravity and capillary pressure, *SIAM J. Sci. Stat. Comput.* 10 (1989) 227–252.
- [26] V. Ginting, G. Lin, J. Liu, On application of the weak Galerkin finite element method to a two-phase model for subsurface flow, *J. Sci. Comput.*, 66 (2016) 225–239.
- [27] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, 2008.
- [28] P. T. Harker and J.-S. Pang, Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications, *Math. Prog.* 48 (1990) 161–220.
- [29] M. Hintermüller, K. Ito, K. Kunisch, The primal-dual active set strategy as a semismooth Newton method, *SIAM J. Optim.* 13 (2003) 865–888.
- [30] 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.
- [31] F.-N. Hwang, H.-L. Lin, X.-C. Cai, Two-level nonlinear elimination based preconditioners for inexact Newton methods with application in shocked duct flow calculation, *Electron. Trans. Numer. Anal.* 37 (2010) 239–251.
- [32] F.-N. Hwang, Y.-C. Su, X.-C. Cai, A parallel adaptive nonlinear elimination preconditioned inexact Newton method for transonic full potential flow problems, *Comp. Fluids* 110 (2015) 96–107.
- [33] H. Jiang, L. Qi, A new nonsmooth equations approach to nonlinear complementarity problems, *SIAM J. Control Opt.* 35 (1997) 178–193.

- [34] C. Kanzow, Inexact semismooth Newton methods for large-scale complementarity problems, *Opt. Meth. Software* 19 (2004) 309–325.
- [35] D. A. Knoll, D. E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *J. Comput. Phys.* 193 (2004) 357–397.
- [36] J. Kou, S. Sun, A new treatment of capillarity to improve the stability of IMPES two-phase flow formulation, *Comput. Fluids* 39 (2010) 1293–1931.
- [37] 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.
- [38] S. Lacroix, Y. V. Vassilevski, J. A. Wheeler, M. F. Wheeler, Iterative solution methods for modeling multiphase flow in porous media fully implicitly, *SIAM J. Sci. Comput.* 25 (2003) 905–926.
- [39] P. J. Lanzkron, D. J. Rose, J. T. Wilkes, An analysis of approximate nonlinear elimination, *SIAM, J. Sci. Comput.* 17 (1996) 538–559.
- [40] K. Lie, S. Krogstad, I. Lagaarden, J. Natvig, H. Nilsen, B. Skaflestad, Open-source MATLAB implementation of consistent discretisations on complex grids, *Comput. Geosci.*, 16 (2012), pp. 297–322.
- [41] L. Liu, D. E. Keyes, Field-split preconditioned inexact Newton algorithms, *SIAM J. Sci. Comput.* 37 (2015) A1388–A1409.
- [42] B. Lu, Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media, PhD dissertation, The University of Texas at Austin, 2008.
- [43] B. Lu, M. F. Wheeler, Iterative coupling reservoir simulation on high performance computers, *Pet. Sci.* 6 (2009) 43–50.
- [44] 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. Meth. Engng.* 69 (2007) 698–728.
- [45] W. A. Mulder, B. Van Leer, Experiments with implicit upwind methods for the Euler equations, *J. Comput. Phys.* 59 (1985) 232–246.
- [46] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, C. Kanzow, The semi-smooth algorithm for large scale complementarity problems, *INFORMS J. Comput.* 13 (2001) 294–311.
- [47] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Math. Oper. Research* 18 (1993) 227–244.
- [48] J. F. Rodrigues, *Obstacle Problems in Mathematical Physics*, North-Holland, Amsterdam, 1987.
- [49] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, Second ed., 2003.
- [50] B. Smith, P. Bjørstad, W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
- [51] J. O. Skogestad, E. Keilegavlen, J.M. Nordbotten, Domain decomposition strategies for nonlinear flow problems in porous media, *J. Comput. Phys.* 234 (2013) 439–451.
- [52] S. Sun, D. E. Keyes, L. Liu, Fully implicit two-phase reservoir simulation with the additive schwarz preconditioned inexact Newton method, *SPE Reservoir Characterization and Simulation Conference and Exhibition*. Society of Petroleum Engineers, 2013.
- [53] T. B. Tan, N. Kaioerakis, A fully implicit, three-dimensional, three-phase simulator with automatic history-matching capability, presented at the 11th SPE Symposium on Reservoir Simulation, Anaheim, CA. Feb. 1991, SPE 21205.
- [54] A. Toselli, O. Widlund, *Domain Decomposition Methods-Algorithms and Theory*, Springer, Berlin, 2005.
- [55] M. Ulbrich, Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems, *SIAM J. Opt.* 11 (2001) 889-917.
- [56] H. Yang, X.-C. Cai, Parallel two-grid semismooth Newton–Krylov–Schwarz method for nonlinear complementarity problems, *J. Sci. Comput.* 47 (2011) 258–280.
- [57] H. Yang, C. Yang, X.-C. Cai, Parallel domain decomposition methods with mixed order discretization for fully implicit solution of tracer transport problems on the cubed-sphere, *J. Sci. Comput.* 61 (2014) 258–280.
- [58] L. C. Young, R. E. Stephenson, A generalized compositional approach for reservoir simulation, *SPE J.*

23 (1983) 727–742.

ACCEPTED MANUSCRIPT