ICES REPORT 17-22

September 2017

An Approximate Jacobian Nonlinear Solver for Multiphase Flow and Transport

by

Gurpreet Singh, Gergina Pencheva, and Mary F. Wheeler

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Gurpreet Singh, Gergina Pencheva, and Mary F. Wheeler, "An Approximate Jacobian Nonlinear Solver for Multiphase Flow and Transport," ICES REPORT 17-22, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, September 2017.

An Approximate Jacobian Nonlinear Solver for Multiphase Flow and Transport

September 10, 2017

Abstract

We present an approximate Jacobian approach for solving nonlinear, multiphase flow and transport problems in porous media. A backward Euler time discretization scheme is used; prior to spatial discretization with a lowest order mixed finite element method (MFEM). This results in a fully implicit nonlinear algebraic system of equations. Conventionally, an exact Jacobian construction is employed during the Newton linearization to obtain a linear system of equations after spatial and temporal discretization. This fully coupled, monolithic linear system; usually in pressure and saturation (or concentration) unknowns, requires specialized preconditioners such as compressed pressure residual (CPR) or two stage preconditioner. These preconditioners operate on the linear system to decouple pressure and saturation (or concentration) degrees of freedom (DOF) in order to use existing linear solvers for positive definite (PD) matrices such as GMRES, AMG etc. can be used. In this work, we present an alternative to two-stage preconditioning (or CPR) for solving the aforementioned monolithic system after Newton linearization. This method relies upon an approximation in the non-linear, fully discrete, variational formulation resulting in decoupling of the DOFs and consequent approximate Jacobian construction. The resulting linear system is easily reduced to a system in pressure DOF circumventing the need for these specialized preconditioners. Further, the linear system has lesser DOF owing to the elimination of saturation (or concentration) unknowns. This nonlinear solver is demonstrated to be as accurate as the exact Jacobian approach, measured in terms of convergence of nonlinear residual to a desired tolerance for both methods. Our numerical results indicate a consistent computational speedup by a factor of approximately 1.32 to 4.0 for the two-phase flow model formulation under consideration. This is related to the DOF of the linear systems for the approximate and exact Jacobian approaches. For multicomponent flow and transport this speedup is expected to be directly proportional to the number of concentration degrees of freedom. A number of field scale numerical simulations are also presented to demonstrate the efficacy of this approach for realistic problems.

Keywords. nonlinear solver, Jacobian approximation, mixed finite elements, cell-centered finite difference, fully-implicit

1 Introduction

Nonlinear processes are a common occurrence for multiphase, flow and transport problems in sub-surface porous media. These nonlinearities originate from empirical, phenomenological and often mechanistic considerations during the modeling of the physical processes. Conventionally, after spatial and temporal discretization of the partial differential equations of the associated model formulation, linearization is necessary to solve the resulting nonlinear system of algebraic equations. The Newton's method is a well-known nonlinear solver and has proven quadratic rate of convergence in the vicinity of the solution assuming a Lipschitz continuity argument. There is an extensive literature dedicated to different modifications of Newton's method that address a wide array of issues in solving nonlinear system of equations (inexact, chord, quasi-Newton, semismooth, etc.), see e.g. [4, 10, 25, 15, 11, 12] and references therein. The semi-smooth Newton methods, for example, are used for treating system of non-smooth equations with lower regularity arising from model inequality constraints, inadvertent roughness due to empirical considerations, or other modeling inconsistencies. This latter inconsistency is often seen in realistic subsurface reservoirs with heterogeneous rock properties, where an empirical description of capillary pressure and relative permeability introduces discontinuities at the interface between different rock types.

For flow and transport problems in porous media, a backward Euler discretization in time with an appropriate spatial discretization gives rise to a fully-implicit nonlinear system of algebraic equations. A nonlinear solver; as described before, is then used to obtain spatial distributions of desired unknowns (pressure, saturation, concentration etc.) at a given time. In doing so, the solution available at the previous time, the time-step size, and the nonlinear solver convergence rates are inherently tied. The primary focus in the development of computationally efficient, nonlinear solution strategies is to either increase the convergence rates (optimally quadratic) or increase time-step sizes although at the cost of additional numerical diffusion. However, these two desirable properties are closely related due to the linearization assumption inherent to the nonlinear solver. A large time-step size requirement strains this linearization assumption; since an initial estimate from previous time solutions is not sufficiently close to the final solution, resulting in reduced convergence rates. Similarly, a requirement on the convergence rate constraints the time-step size. Since a rigorous derivation relating time-step size and nonlinear convergence rate is not often achievable in the light of the wide ranging model nonlinearities, we draw our conclusions based upon observed numerics.

Several approaches have been proposed that aim to alleviate or circumvent some of these aforementioned issues in favor of overall computational efficiency. A number of these approaches [6, 34, 18, 14] improve nonlinear convergence rates; for a given time-step size, that rely upon modifying the Newton step-size or descent direction or both. These modifications are either based upon determining an optimal nonlinear step size using line-search algorithms or altering descent direction (Jacobian) using prior knowledge of the regularity of functions contributing to the Jacobian. The simplest example of modifying the descent direction occurs in the case of slightly compressible fluid description where the contribution of the density derivative with respect to pressure to the Jacobian is often considered negligible. This latter modification improves the overall efficiency by neglecting the computationally expensive evaluation of the density derivative. Another such approach, is the reduced Newton algorithm proposed by [18, 14] that relies upon saturation or concentration updates in an order determined by pressure potential. The authors report an overall reduction in computational cost due to larger time-step sizes for which the nonlinear iterations converge. However, the sequential nature of potential reordering and consequent saturation/concentration updates in this proposed nonlinear solution algorithm might pose parallel scalability issues.

In this work, we present a nonlinear solver based upon modifying the Newton descent direction to improve the overall computational efficiency for numerical reservoir simulations. Although, this work has been extended for fully implicit, multiphase, compositional flow [30], the model complexity precludes a fair comparison between the proposed and conventional approaches. We therefore restrict ourselves to a slightly compressible, two phase flow model for the sake of simplicity. This allows us to compare the differences between the conventional Newton method (exact Jacobian) and our approach (approximate Jacobian) both in terms of nonlinear system formulation and consequent numerical benchmarking. We begin with a description of a two-phase flow model formulation followed by a brief discussion of a mixed finite element spatial discretization and its relation to the well known cell-centered finite difference. We then present a δ notation relying upon Gâteaux derivatives to linearize the semi-discrete nonlinear partial differential equations obtained after temporal discretization prior to spatial discretization. This is followed by a detailed discussion on the conventional and proposed approaches. We use this aforementioned δ notation to simplify the description and easily distinguish the differences in Jacobian construction for the two approaches. Next we describe the nonlinear solvers and preconditioners used in this work with a brief discussion on the accuracy of the two nonlinear solvers. Finally, we present extensive numerical experiments for benchmarking the proposed approach against the conventional method to determine the computational speedup and overall increase in efficiency.

2 Model Formulation

We begin by describing the model formulation for immiscible, two-phase, slightly compressible flow in a porous medium which is widely accepted in several porous media communities such as oil and gas, ground water hydrology and environmental engineering.

2.1 Phase Conservation Equations

Consider a time interval $(0, T]$, along with a spatial domain, $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3 with boundary $\partial\Omega$ and outward unit normal $\boldsymbol{n}$. The mass conservation equation for phase α is

$$\frac{\partial \left(\phi \rho_\alpha S_\alpha\right)}{\partial t} + \nabla\cdot \boldsymbol{u}_\alpha = q_\alpha \text{ in } \Omega \times (0, T], \tag{1}$$

where ϕ is the rock porosity and ρ_α, S_α, u_α and q_α are density, saturation, velocity and source/sink term, respectively of phase α. The Darcy velocity is given by,

$$\boldsymbol{u}_\alpha = -K \rho_\alpha \frac{k_{r\alpha}}{\mu_\alpha} \left(\nabla p_\alpha - \rho_\alpha \boldsymbol{g}\right) \text{ in } \Omega \times (0, T] \tag{2}$$

Here, K and $\boldsymbol{g}$ are the rock permeability and gravitational constant, respectively. Further, $k_{r\alpha}$, μ_α and p_α are the relative permeability, viscosity and pressure of phase α.

2.2 Initial and Boundary Conditions

Although not restrictive, for the sake of simplicity we assume no flow boundary conditions.

$$\boldsymbol{u}_\alpha \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \times (0, T] \tag{3}$$

$$p_\alpha = p_\alpha^0, \quad S_\alpha = S_\alpha^0, \text{ at } \Omega \times \{t = 0\} \tag{4}$$

Here, p_α^0, S_α^0 are the initial conditions for pressure and saturation of phase α.

2.3 Constraints and Other Conditions

The phase saturations S_α are constrained as,

$$\sum_\alpha S_\alpha = 1. \tag{5}$$

We assume capillary pressure and relative permeabilities to be continuous and monotonic functions of phase saturations. A precise description of the functional forms is avoided to maintain generality.

$$p_c = f(S_o) = p_w - p_o \tag{6}$$

$$k_{r\alpha} = k_{r\alpha}(S_\alpha) \tag{7}$$

Further, both oil and water phase are assumed to slightly compressible with phase densities evaluated using

$$\rho_\alpha = \rho_{\alpha,ref} \exp\left[c_{f\alpha}(p_\alpha - p_{\alpha,ref})\right]. \tag{8}$$

Here, $c_{f\alpha}$ is the compressibility and $\rho_{\alpha,ref}$ is the density of phase α at the reference pressure $p_{\alpha,ref}$.

3 MFEM as CCFD

The cell-centered finite difference scheme (CCFD) is well known in flow and transport in porous medium communities. In this work, we use a mixed finite element method (MFEM) for spatial discretization related to the CCFD scheme in an effort to assist code portability for legacy reservoir simulators. Earlier works [28] show that the mixed finite element method with the lowest order Raviart-Thomas-Nedelec (RTN) spaces [26, 23] on a rectangular grid reduces to a cell-centered finite difference scheme due to the choice of a numerical quadrature rule. The MFEM therefore results in the same nonlinear system; after fluxes are eliminated, as for the CCFD scheme, making the two schemes equivalent. Some results on superconvergence of this MFEM or CCFD scheme with flux boundary conditions for an elliptic problem can be found in [22]. An expanded MFEM to handle full tensor permeability and general geometry using logically rectangular grids was presented in [2, 3]. Later this scheme was applied to non-linear, two-phase and black-oil flow problems by [9] and [24], respectively. Here an auxiliary flux term is introduced to avoid inversion of zero relative permeabilities. However, for discontinuous permeability tensors the gradient in pressure and hence the auxiliary flux is also discontinuous wherein a hybrid form of the expanded mixed method is necessary to recover accuracy.

A multipoint flux mixed finite element (MFMFE) method was later presented by [33] to handle non-smooth grids and permeabilities. This scheme avoids Lagrange pressure multipliers at the discontinuous interfaces introduced by the former expanded MFEM while preserving the cell-centered structure of the CCFD scheme. The MFMFE scheme can also handle full tensor permeabilities on logically rectangular distorted hexahedra to capture general geometries. An expanded form of the MFMFE scheme is then used to avoid inversion of zero relative permeabilities in the case of non-linear multiphase flow problems. This MFMFE scheme has been extended to handle complex, non-linear flow and transport problems including equation of state compositional flow [30]. In this work, we use this latter MFE scheme for the sake of generalization to more complex flow models and simplicity in outlining the differences between approximate and exact Jacobian appraoches. In what follows, a two-phase, slightly compressible flow model is used resulting in a nonlinear algebraic system of equations. Further details regarding this MFE discretization applied to a two phase flow model formulation can be found in Appendix A.

4 The δ Notation

Conventionally, to solve a nonlinear partial differential equation (PDE) system using finite element methods, one begins with a fully-discrete weak formulation. This is followed by a linearization step on the resulting nonlinear system of algebraic equations. In contrast, for the approximate Jacobian approach presented here, Newton linearization is performed on the discrete-in-time variational formulation before spatial discretization is considered. We use the δ notation when calculating the required derivatives which allows for simplicity of describing the approximations made in aforementioned method. Although the conventional method can be used, we consider the latter for the sake of simplicity in comparing the two formulations. To solve the nonlinear equation $\mathcal{F}(p) = 0$, Newton method solves a sequence of linear problems,

$$\mathcal{F}'(p^k)\delta p^k = -\mathcal{F}(p^k),$$

for $\delta p^k = p^{k+1} - p^k$, the difference between two consecutive iterates. Here, $\mathcal{F}'(p)$ is the Fréchet derivative of the operator $\mathcal{F}$. To show the similarities between computing derivatives of (nonlinear) operators and derivatives of real-valued functions, note that $\mathcal{F}'(p)$ is also the Gâteaux derivative of $\mathcal{F}$ [21]. The latter is a generalization of the classical directional derivative and is defined as

$$\mathcal{F}'(p)(\delta p) := \lim_{\varepsilon \to 0} \frac{\mathcal{F}(p + \varepsilon \delta p) - \mathcal{F}(p)}{\varepsilon}. \tag{9}$$

We formally write δ as an operator that maps $\mathcal{F}(p)$ to its Gâteaux derivative in the direction δp, i.e.

$$\delta(\mathcal{F}(p)) := \mathcal{F}'(p)(\delta p). \tag{10}$$

Note that the δ operator is independent of the spatial operators ∇ and $\nabla \cdot$. Please see Appendix B for an example to calculate the Gâteaux derivative.

5 Fully Implicit with Exact Jacobian Construction

We first describe a fully implicit scheme for solving the two-phase flow model formulation discussed earlier. As described in Section 3, an RTN_0 space is used for spatial discretization with a specific quadrature rule which results in the well-known finite difference scheme. A backward Euler scheme is used for temporal discretization resulting in a fully-implicit, nonlinear system of equations. The exact Jacobian construction refers to the Newton linearization of this nonlinear system of equations where the derivatives of the nonlinear terms with respect to the primary unknowns are evaluated exactly without making any approximations. We later describe an inexact Jacobian construction wherein approximations are made to facilitate easier elimination of the some of the primary unknowns to derive a reduced linear system. We select oil phase concentration, c_o (defined below), oil phase pressure, p_o, phase velocities, $\boldsymbol{u}_\alpha$, and auxiliary phase velocities $\tilde{\boldsymbol{u}}_\alpha$ as the primary unknowns following by temporal and spatial discretization prior to Newton linearization of the resulting system. Please note that the phase velocities are taken to be primary unknowns to draw out some of the necessary details and are eliminated after Newton linearization resulting in a reduced system with p_o and c_o as the unknowns. We suggest that the reader to go over this section cursorily, returning only to compare the differences between exact and inexact Jacobian constructions after linearization, as described in Sections 5.2 and 6.2, respectively.

5.1 Time Discrete Variational Formulation

Let us define c_α as the concentration of phase α as follows,

$$c_\alpha = \rho_\alpha S_\alpha. \tag{11}$$

Then using backward Euler scheme for time discretization of the two phase mass conservation equations results in an implicit system in pressure (p_o), concentration (c_o) and velocities (u_o, u_w) unknowns.

$$\frac{(\phi c_\alpha)^{n+1} - (\phi c_\alpha)^n}{\Delta t} + \nabla\cdot\boldsymbol{u}_\alpha^{n+1} = q_\alpha^{n+1}, \tag{12a}$$

$$\boldsymbol{u}_o^{n+1} = \left(-K\rho_o\frac{k_{ro}}{\mu_o}\left(\nabla p_o - \rho_o\boldsymbol{g}\right)\right)^{n+1}, \tag{12b}$$

$$\boldsymbol{u}_w^{n+1} = \left(-K\rho_w\frac{k_{rw}}{\mu_w}\left(\nabla(p_0 - p_c) - \rho_w\boldsymbol{g}\right)\right)^{n+1}. \tag{12c}$$

The saturation constraint, Eqn. (5), can be expressed in terms of the unknowns c_α and p_o using Eqn. (11) as,

$$\sum_\alpha \frac{c_\alpha}{\rho_\alpha} = 1. \tag{12d}$$

The phase densities (ρ_α) are explicit functions of p_α whereas the capillary pressure (p_c) is an explicit function of water saturation and can be easily evaluated. To simplify notation we suppress time index $n+1$ unless superscript n is explicitly used to denote known quantities at the previous time step.

$$\frac{\left(\phi\rho_w\left(1-\frac{c_o}{\rho_o}\right)\right)-\left(\phi\rho_w\left(1-\frac{c_o}{\rho_o}\right)\right)^n}{\Delta t}+\nabla\cdot\boldsymbol{u}_w=q_w \tag{13a}$$

$$\frac{(\phi c_o)-(\phi c_o)^n}{\Delta t}+\nabla\cdot\boldsymbol{u}_o=q_o \tag{13b}$$

$$\boldsymbol{u}_o=-K\lambda_o\left(\nabla p_o-\rho_o\boldsymbol{g}\right) \tag{13c}$$

$$\boldsymbol{u}_w=-K\lambda_w\left(\nabla(p_0-p_c)-\rho_w\boldsymbol{g}\right) \tag{13d}$$

$$\lambda_\alpha=\lambda_\alpha(p_o,c_o)=\rho_\alpha(p_o)\frac{k_{r\alpha}(c_o,p_o)}{\mu_\alpha} \tag{13e}$$

$$p_c=p_c(S_o)=p_c(c_o,p_o) \tag{13f}$$

In order to avoid inverting a zero; water and oil relative permeabilities at irreducible and residual saturations, respectively, we define an auxiliary velocity unknown $\tilde{\boldsymbol{u}}_\alpha$ and rewrite the constitutive equations or Darcy's law for each phase α as,

$$\boldsymbol{u}_\alpha=\lambda_\alpha\tilde{\boldsymbol{u}}_\alpha,\qquad \tilde{\boldsymbol{u}}_\alpha=-K\left(\nabla p_\alpha-\rho_\alpha\boldsymbol{g}\right). \tag{14}$$

We define velocity and pressure spaces as $\boldsymbol{V}=\{\boldsymbol{v}\text{ in }H(\text{div};\Omega):\boldsymbol{v}\cdot\boldsymbol{n}=0\text{ on }\partial\Omega\}$ and $W\equiv L^2(\Omega)$, respectively. The expanded mixed variational problem is: Given c_o^n and p_o^n find $p_o\in W$, $\boldsymbol{u}_\alpha\in V$ and $c_o\in W$ such that,

$$\left(\frac{(\phi c_o)-(\phi c_o)^n}{\Delta t},w\right)+(\nabla\cdot\boldsymbol{u}_o,w)=(q_o,w) \tag{15a}$$

$$\left(\frac{\phi\rho_w\left(1-\frac{c_o}{\rho_o}\right)-\left(\phi\rho_w\left(1-\frac{c_o}{\rho_o}\right)\right)^n}{\Delta t},w\right)+(\nabla\cdot\boldsymbol{u}_w,w)=(q_w,w) \tag{15b}$$

$$\left(K^{-1}\tilde{\boldsymbol{u}}_o,\boldsymbol{v}\right)-(p_o,\nabla\cdot\boldsymbol{v})=(\rho_o\boldsymbol{g},\boldsymbol{v}) \tag{15c}$$

$$\left(K^{-1}\tilde{\boldsymbol{u}}_w,\boldsymbol{v}\right)-(p_o,\nabla\cdot\boldsymbol{v})=-(p_c,\nabla\cdot\boldsymbol{v})+(\rho_w\boldsymbol{g},\boldsymbol{v}) \tag{15d}$$

$$(\boldsymbol{u}_\alpha,\boldsymbol{v})=(\lambda_\alpha\tilde{\boldsymbol{u}}_\alpha,\boldsymbol{v}) \tag{15e}$$

Here, $w\in W$ and $v\in\boldsymbol{V}$.

5.2 Linearization

Next we linearize the above system to construct an exact Jacobian. We omit the nonlinear /Newton iteration counter; represented by the superscript k, for simplicity of description in the following. The Newton step δa^k is then defined as,

$$\delta a^k = a^{k+1} - a^k. \tag{16}$$

$$(\phi\delta c_o, w) + (\nabla\cdot\delta\boldsymbol{u}_o, w)\,\Delta t = -\left((\phi c_o) - (\phi c_o)^n, w\right) - \Big[(\nabla\cdot\boldsymbol{u}_o, w) - (q_o, w)\Big]\Delta t \tag{17a}$$

$$\begin{aligned}\left(\phi\Big(-\frac{\rho_w}{\rho_o}\delta c_o + c_{fw}\rho_w(1-\frac{c_o}{\rho_o})\delta p_o + \rho_w c_{fo}\frac{c_o}{\rho_o}\delta p_o\Big), w\right) + (\nabla\cdot\delta\boldsymbol{u}_w, w)\,\Delta t \\ = -\Bigg[\frac{1}{\Delta t}\left(\phi\rho_w\left(1-\frac{c_o}{\rho_o}\right) - \phi\rho_w\left(1-\frac{c_o}{\rho_o}\right)^n, w\right) \\ + (\nabla\cdot\boldsymbol{u}_w, w) - (q_w, w)\Bigg]\end{aligned} \tag{17b}$$

$$\begin{aligned}\left(K^{-1}\delta\tilde{\boldsymbol{u}}_o, \boldsymbol{v}\right) - (\delta p_o, \nabla\cdot\boldsymbol{v}) - (c_{fo}\rho_o\boldsymbol{g}\delta p_o, \boldsymbol{v}) \\ = -\left[\left(K^{-1}\tilde{\boldsymbol{u}}_o, \boldsymbol{v}\right) - (p_o, \nabla\cdot\boldsymbol{v}) - (\rho_o\boldsymbol{g}, \boldsymbol{v})\right]\end{aligned} \tag{17c}$$

$$\begin{aligned}\left(K^{-1}\delta\tilde{\boldsymbol{u}}_w, \boldsymbol{v}\right) - (\delta p_o, \nabla\cdot\boldsymbol{v}) - \left(c_{fw}\rho_w\left[\delta p_o - \frac{\partial p_c}{\partial c_o}\delta c_o\right]\boldsymbol{g}, \boldsymbol{v}\right) + \left(\frac{\partial p_c}{\partial c_o}\delta c_o + \frac{\partial p_c}{\partial p_o}\delta p_o, \nabla\cdot\boldsymbol{v}\right) \\ = -\left[\left(K^{-1}\tilde{\boldsymbol{u}}_w, v\right) - (p_o, \nabla\cdot\boldsymbol{v}) + (p_c, \nabla\cdot\boldsymbol{v}) - (\rho_w\boldsymbol{g}, \boldsymbol{v})\right]\end{aligned} \tag{17d}$$

$$\begin{aligned}(\delta\boldsymbol{u}_\alpha, \boldsymbol{v}) - (\lambda_\alpha\delta\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}) - \left(\frac{\partial\lambda_\alpha}{\partial p_o}\delta p_o\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}\right) - \left(\frac{\partial\lambda_\alpha}{\partial c_o}\delta c_o\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}\right) \\ = -\Big((\boldsymbol{u}_\alpha, \boldsymbol{v}) - (\lambda_\alpha\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v})\Big)\end{aligned} \tag{17e}$$

For slightly compressible flow models, the contribution of the third term in each of the Eqns. (17c), (17d), and (17e) is small and is often neglected. We assume that the capillary pressure at the domain boundaries is zero or,

$$\int_{\partial\Omega} p_c\,\boldsymbol{v}\cdot\boldsymbol{n} = 0, \tag{18}$$

in the above equations. This is consistent with the physical observation of capillary end effects in core-flooding experiments due to absence of capillary pressure at the core boundaries. Another argument in favor of this assumption comes from subsurface porous media considerations where saturation condition and hence capillary pressure at the domain boundaries are usually not known.

$$\begin{bmatrix} A & B & 0 & 0 \\ 0 & 0 & D_o & B^T \\ \Lambda_o & \Lambda_{op} & \Lambda_{oc} & I \end{bmatrix} \begin{bmatrix} \delta\tilde{\boldsymbol{u}}_o \\ \delta p_o \\ \delta c_o \\ \delta\boldsymbol{u}_o \end{bmatrix} = \begin{bmatrix} -R_{1o} \\ -R_{2o} \\ -R_{3o} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} A & B & 0 & 0 \\ 0 & C_w & D_w & B^T \\ \Lambda_w & \Lambda_{wp} & \Lambda_{wc} & I \end{bmatrix} \begin{bmatrix} \delta\tilde{\boldsymbol{u}}_w \\ \delta p_o \\ \delta c_o \\ \delta\boldsymbol{u}_w \end{bmatrix} = \begin{bmatrix} -R_{1w} \\ -R_{2w} \\ -R_{3w} \end{bmatrix} \tag{20}$$

$$\Big(B^T\Lambda_o A^{-1}B - B^T\Lambda_{op}\Big)\delta p_o + \Big(D_o - B^T\Lambda_{oc}\Big)\delta c_o = B^T\Big(R_{3o} - \Lambda_o A^{-1}R_{1o}\Big) - R_{2o} \tag{21}$$

$$\Big(C_w + B^T\Lambda_w A^{-1}B - B^T\Lambda_{wp}\Big)\delta p_o + \Big(D_w - B^T\Lambda_{wc}\Big)\delta c_o = B^T\Big(R_{3w} - \Lambda_w A^{-1}R_{1w}\Big) - R_{2w} \tag{22}$$

Eqns. (21) and (22) constitute the linear system of equations in unknowns p_o and c_o. The choice of quadrature rules, discussed in previous section, results in different sparsity patterns in matrix A; block diagonal for the MFMFE scheme [33], and diagonal for the CCFD scheme [3]. Similar differences in sparsity patterns also occur in matrices $\Lambda_{\alpha,p/c}$ due to differences in quadrature rules used for evaluating the integral.

$$\begin{aligned}
\Lambda_\alpha &= diag(-\lambda_\alpha) \\
\Lambda_{\alpha a} &= \Big[\frac{\partial\lambda_\alpha}{\partial a}\tilde{\boldsymbol{u}}_\alpha\Big]_{ij}\ ,\ a = p_o \text{ or } c_o \\
C_w &= diag\Big(c_{fw}\rho_w\big(1 - \frac{c_o}{\rho_o}\big) + \rho_w c_{fo}\frac{c_o}{\rho_o}\Big) \\
D_w &= diag\Big(-\phi\,\frac{\rho_w}{\rho_o}\Big) \\
D_o &= diag\big(\phi\big)
\end{aligned} \tag{23}$$

We abuse the notation by reusing subscripts i and j, used differently in the previous sections, to denote matrix entries of $\Lambda_{a\alpha}$.

6 Fully Implicit using Approximate Jacobian Construction

Here, we present an approximate Jacobian construction resulting from the nonlinear system of algebraic equations after spatial and temporal discretizations as described in the previous section. This formulation differs in the choice of primary unknowns which are oil phase pressure, p_o, saturation of both phases, S_o and S_w and phase velocities, u_o and u_w. The saturation constraint is not used to eliminate one of the saturation unknowns and treated as an additional constraint. Again, as described in Section 5, the phase velocities are eliminated here as well after linearization resulting in a reduced linear system. The approximate Jacobian approach is designed to trivially eliminate phase saturations, with negligible computational overheads, resulting in a reduced linear system in oil phase pressure unknown, p_o. In what follows, we first present the temporal and spatial discretizations to arrive at a nonlinear system of algebraic equations. Thereafter, an approximate Jacobian construction is described for linearization and consequently the resulting reduced linear system. One important point to note is that a special two-stage or CPR (Compressed Pressure Residual) preconditioner is not needed for this linear system in contrast to the exact Jacobian construction. The numerical results presented in Section 8 later demonstrate the computational savings owing to the reduced linear system inherent to the approximate Jacobian construction when compared to the exact Jacobian construction.

6.1 Time Discrete Variational Formulation

We use backward Euler scheme for time discretization of the two phase mass conservation equations resulting in an implicit system in pressure (p_o), saturations (S_o, S_w) and velocities (u_o, u_w) unknowns. Again, ρ_α is an explicit function of p_α and can be easily evaluated whereas p_c is an explicit function of saturation. As before, to simplify notation we suppress time index $n+1$ unless superscript n is explicitly used to denote known quantities at the previous time step. The time-discrete form is then given by,

$$\frac{(\phi\rho_\alpha S_\alpha) - (\phi\rho_\alpha S_\alpha)^n}{\Delta t} + \nabla\cdot\boldsymbol{u}_\alpha = q_\alpha \tag{24a}$$

$$\boldsymbol{u}_o = -K\lambda_o\left(\nabla p_o - \rho_o\boldsymbol{g}\right) \tag{24b}$$

$$\boldsymbol{u}_w = -K\lambda_w\left(\nabla(p_0 - p_c) - \rho_w\boldsymbol{g}\right) \tag{24c}$$

$$\lambda_\alpha = \lambda_\alpha(p_o, S_\alpha) = \rho_\alpha\frac{k_{r\alpha}(S_\alpha)}{\mu_\alpha} \tag{24d}$$

Again, the expanded mixed variational problem is: Given S_α^n and p_o^n find $p_o \in W$, $u_\alpha \in V$ and $S_\alpha \in W$ such that,

$$\left(\frac{(\phi\rho_\alpha S_\alpha) - (\phi\rho_\alpha S_\alpha)^n}{\Delta t}, w\right) + (\nabla\cdot\boldsymbol{u}_\alpha, w) = (q_\alpha, w) \tag{25a}$$

$$\left(K^{-1}\tilde{\boldsymbol{u}}_o, \boldsymbol{v}\right) - (p_o, \nabla\cdot\boldsymbol{v}) = (\rho_o\boldsymbol{g}, \boldsymbol{v}) \tag{25b}$$

$$\left(K^{-1}\tilde{\boldsymbol{u}}_w, \boldsymbol{v}\right) - (p_o, \nabla\cdot\boldsymbol{v}) = -\left(p_c, \nabla\cdot\boldsymbol{v}\right) + (\rho_w\boldsymbol{g}, \boldsymbol{v}) \tag{25c}$$

$$(\boldsymbol{u}_\alpha, \boldsymbol{v}) = (\lambda_\alpha\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}) \tag{25d}$$

$$\sum_\alpha S_\alpha = 1 \tag{25e}$$

As before, $w \in W$ and $v \in V$.

6.2 Linearization

A variant of the chord method is,

$$a^{k+1} = a^k - (A^k)^{-1} F(a^k) \tag{26}$$

where $A^k \approx F'(x^k)$ and can also be viewed as a preconditioned nonlinear Richardson iteration. For more details regarding chord method or its variants along with convergence rates please read [15].

$$\begin{aligned}(\phi\rho_\alpha \delta S_\alpha, w) + (\phi c_{f\alpha}\rho_\alpha S_\alpha \delta p_o, w) + (\nabla\cdot\delta\boldsymbol{u}_\alpha, w)\,\Delta t \\ = -\left((\phi\rho_\alpha S_\alpha) - (\phi\rho_\alpha S_\alpha)^n, w\right) - \left[(\nabla\cdot\boldsymbol{u}_\alpha, w) - (q_\alpha, w)\right]\Delta t\end{aligned} \tag{27a}$$

$$\begin{aligned}\left(K^{-1}\delta\tilde{\boldsymbol{u}}_o\right) - (\delta p_o, \nabla\cdot\boldsymbol{v}) - (c_{fo}\rho_o\boldsymbol{g}\delta p_o, \boldsymbol{v}) \\ = -\left[\left(K^{-1}\tilde{\boldsymbol{u}}_o, \boldsymbol{v}\right) - (p_o, \nabla\cdot\boldsymbol{v}) - (\rho_o\boldsymbol{g}, \boldsymbol{v})\right]\end{aligned} \tag{27b}$$

$$\begin{aligned}\left(K^{-1}\delta\tilde{\boldsymbol{u}}_w, v\right) - (\delta p_o, \nabla\cdot\boldsymbol{v}) - \left(c_{fw}\rho_w\left[\delta p_o - \frac{\partial p_c}{\partial S_w}\delta S_w\right]\boldsymbol{g}, \boldsymbol{v}\right) + \left(\frac{\partial p_c}{\partial S_w}\delta S_w + \frac{\partial p_c}{\partial p_o}\delta p_o, \nabla\cdot\boldsymbol{v}\right) \\ = -\left[\left(K^{-1}\tilde{\boldsymbol{u}}_w, v\right) - (p_o, \nabla\cdot\boldsymbol{v}) + (p_c, \nabla\cdot\boldsymbol{v}) - (\rho_w\boldsymbol{g}, \boldsymbol{v})\right]\end{aligned} \tag{27c}$$

$$\begin{aligned}(\delta\boldsymbol{u}_\alpha, \boldsymbol{v}) - (\lambda_\alpha(p_o, S_\alpha)\,\delta\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}) - \left(\frac{\partial\lambda_\alpha}{\partial p_o}\delta p_o\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}\right) - \left(\frac{\partial\lambda_\alpha}{\partial S_\alpha}\delta S_\alpha\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}\right) \\ = -\left[(\boldsymbol{u}_\alpha, v) - (\lambda_\alpha(p_o, S_\alpha)\,\tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v})\right]\end{aligned} \tag{27d}$$

$$\sum_\alpha \delta S_\alpha = -\left[\sum_\alpha S_\alpha - 1\right] \tag{27e}$$

The above constitute 7 linear equations in 7 unknowns (p_o, S_α, $\boldsymbol{u}_\alpha$, $\tilde{\boldsymbol{u}}_\alpha$). We neglect the third term in each of the Eqns. (27b) thru (27d) as in the case of exact Jacobian construction in the previous section. Additionally, we also neglect the fourth term in each of the Eqns. (27c) and (27d) to construct our approximate Jacobian. These additional approximations allow us to eliminate the saturation unknowns (S_α) in favor of the pressure unknowns (p_o) as shown below. In matrix form this linear system can be expressed as,

$$\begin{bmatrix} A & B & 0 & 0 \\ 0 & \widehat{C}_\alpha & \widehat{D}_\alpha & B^T \\ \Lambda_\alpha & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \delta\tilde{\boldsymbol{u}}_\alpha \\ \delta p_o \\ \delta S_\alpha \\ \delta\boldsymbol{u}_\alpha \end{bmatrix} = \begin{bmatrix} -\widehat{R}_{1\alpha} \\ -\widehat{R}_{2\alpha} \\ -\widehat{R}_{3\alpha} \end{bmatrix}. \tag{28}$$

Eliminating $\tilde{\boldsymbol{u}}_\alpha$, $\boldsymbol{u}_\alpha$ from Eqns. (27a) using Eqns. (27b), (27c) and (27d) we obtain a system of linear equations in p_o and S_α.

$$\begin{aligned}\Lambda_\alpha &= diag(-\lambda_\alpha) \\ \widehat{C}_\alpha &= diag\left(\phi c_{f\alpha}\rho_\alpha S_\alpha\right) \\ \widehat{D}_\alpha &= diag\left(\phi\rho_\alpha\right)\end{aligned} \tag{29}$$

$$(\widehat{C}_\alpha + B^T \Lambda_\alpha A^{-1} B)\delta p_o + \widehat{D}_\alpha \delta S_\alpha = B^T\left(\widehat{R}_{3\alpha} - \Lambda_\alpha A^{-1}\widehat{R}_{1\alpha}\right) - R_{2\alpha} \tag{30}$$

$$\delta S_\alpha = \left(\widehat{D}_\alpha\right)^{-1}\left(B^T\left(\widehat{R}_{3\alpha} - \Lambda_\alpha A^{-1}\widehat{R}_{1\alpha}\right) - \widehat{R}_{2\alpha} - \left(\widehat{C}_\alpha + B^T\Lambda_\alpha A^{-1}B\right)\delta p_o\right) \tag{31}$$

Eqns. (30) and (31) are two equations in three unknowns δp_o, δS_o and, δS_w. Since D_α is a diagonal matrix, it can be easily inverted to express δS_α in terms of δp_o. Substituting these δS_α in Eqn. (27e), we obtain a further reduced linear system of equations in δp_o only.

$$\begin{aligned}\sum_\alpha \left(\widehat{D}_\alpha\right)^{-1}\left(\widehat{C}_\alpha + B^T\Lambda_\alpha A^{-1}B\right)\delta p_o = &\left[\sum_\alpha S_\alpha - 1\right] \\ &+ \sum_\alpha \left(\widehat{D}_\alpha\right)^{-1}\left(B^T\left(\widehat{R}_{3\alpha} - \Lambda_\alpha A^{-1}\widehat{R}_{1\alpha}\right) - \widehat{R}_{2\alpha}\right)\end{aligned} \tag{32}$$

The nonlinear residuals (R_α) for the exact (Eqns. (21) and (22)), and approximate Jacobian (Eqns. (30)) approaches converge to the same solution. The convergence criteria is kept the same for the two approaches and requires that a max norm of the phase residuals R_α is less than a desired tolerance ϵ. Hence there is no loss of accuracy, in terms of mass conservation, due to the aforementioned approximation. The only difference is that the time step sizes; for nonlinear convergence, are usually larger for the former than the latter. Furthermore, the approximate Jacobian requires more number of nonlinear iterations when compared to the exact approach. However, the computational times of the former remain lower than the latter approach.

7 Linear Solvers and Preconditioners

The monolithic linear system obtained using the exact Jacobian approach is sparse, highly non-symmetric, ill-conditioned, and contains blocks with different nature. Particularly effective approach to solve it is to use the Generalized Minimum RESidual (GMRES) method [29] with two-stage preconditioning that decouples the pressure and saturation (or concentration) variables for each grid block [32, 16, 9, 20, 19, 7]. There are several decoupling techniques where the matrix is transformed such that the pressure-concentration block (or submatrix) has zero diagonal [20]. The two predominantly used matrix transformations are the Constraint Pressure Reduction (CPR) [32] and the Householder reflection [13].

As mentioned earlier, the CPR or two-stage preconditioner locally decouples pressure and concentration using matrix operations. This results in lower triangular sub-matrices that is reduced to a diagonal matrix assuming the off-diagonal terms after the aforementioned matrix operation are small. The monolithic system is then reduced and solved using conventional linear solvers for definite systems (e.g. GMRES) to obtain an update in the pressure and saturation unknowns. On the other hand, the approximate Jacobian approach makes an approximation at the nonlinear level; during linearization of the algebraic system, prior to the construction of the linear monolithic system. These approximations allow us to diagonalize these sub-matrices at the cost of additional nonlinear iterations. In our numerical experiments, we observe that the computational cost of these additional nonlinear iterations for the approximate approach are consistently less than the costs incurred by the two-stage preconditioner for the exact approach.

For our numerical simulations, for exact Jacobian approach we use two-stage preconditioned GMRES method with Householder reflection [13] to decouple the pressure and concentration variables. We then employ 4 different preconditioners for the pressure block: multilevel incomplete LU (MLILU) [5], algebraic multigrid (AMG) [27, 31], and their two-stage preconditioner versions using the line successive over-relaxation (LSOR) as a smoother in the vertical direction (MLILU2 and AMG2). The last two methods are considered since they balance the major computational cost of the initialization of the pressure multilevel preconditioners. For the concentration block we use a block Gauss–Seidel preconditioner. For more details, see [19]. For the approximate Jacobian approach, we have only pressure variables and the system is positive definite, so there is no need to employ two-stage preconditioner. Here we simply use a GMRES solver with the above 4 preconditioners, MLILU, MLILU2, AMG, and AMG2.

8 Numerical Results

In this section, we present a number of numerical results to benchmark and compare the proposed approximate Jacobian approach against the conventional exact Jacobian method. These numerical experiments are performed using our in-house reservoir simulator IPARS (Integrated Parallel Accurate Reservoir Simulator). A number of user specifications are required for initial, minimum, and maximum time-step sizes during simulation run. The maximum permissible nonlinear iterations; for the nonlinear solve, are taken to be 15. When these iterations are exceeded the time-step size is reduced by a half for the following time-steps. A successful non-linear solve with iterations below the aforementioned iteration threshold results in an augmentation in the time-step size for the next non-linear solve. This is a fairly standard practice in commercial reservoir simulators for computational efficiency. Figure 1 shows a schematic of time-step size reduction and augmentation used in a number of commercial reservoir simulators. Here, β and α are the time-step size reduction and augmentation factors, respectively.

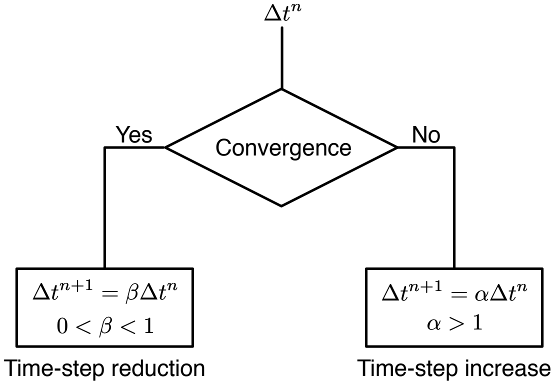

Figure 1: Flow-chart for time step size variation with nonlinear convergence

8.1 Quarter Five Spot Pattern

We present a comparison of CPU runtimes for the two approaches using a quarter five spot well pattern. The computational domain is 80ft×100ft×100ft discretized using 4×50×50 elements. The reservoir properties are taken to be homogeneous with an isotropic, diagonal permeability tensor of 10 mD and a porosity of 0.2. Further, the oil and water phase densities and compressibilities are 56 and 62 lb/ft^3, and 10^{-4} and 10^{-6}, respectively. The gravity vector is taken to be going into the plane of paper in Figure 3 with an initial condition of 1000 psi and 0.2 for the pressure and saturation, respectively. The numerical simulations are performed for a total of 500 days for all of the cases described in this subsection. Figure 2 shows the relative permeability and capillary pressure curves used in this numerical experiment.

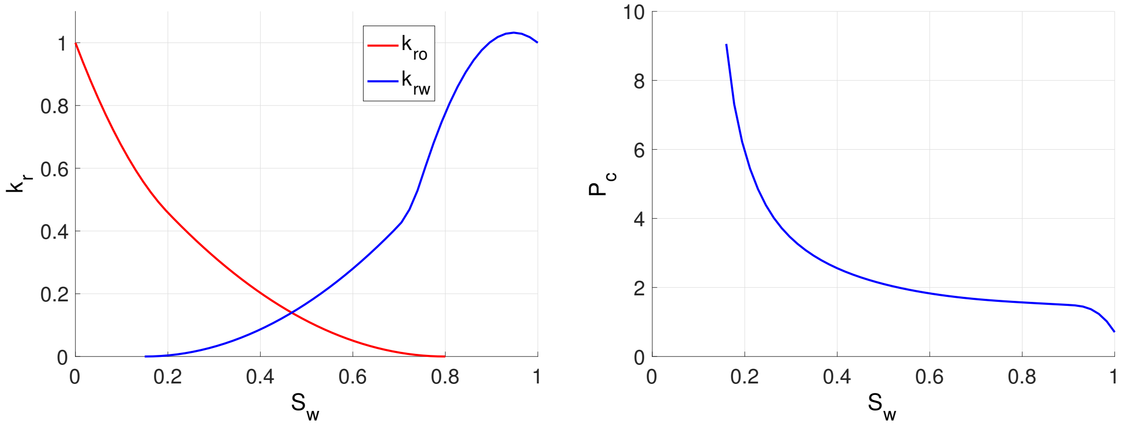

Figure 2: Relative permeability (left) and capillary pressures (right) as function of water saturation for the quarter five spot pattern.

We consider four cases with increasing time-step sizes to identify computational speedups obtained. Case 1 uses a maximum time-step size of 0.2 days for which we observed one or two additional nonlinear iterations for the approximate approach when compared to the exact approach. A larger maximum time-step size of 1.0 day results in substantial increase in additional nonlinear iterations however, the computational speedup is not affected adversely. Please note that Cases 1 and 2 were chosen such that the aforementioned time-step cuts do not occur throughout the simulation run. Table 2 shows the time-step reduction and augmentation factors used for each of the four comparison cases. The check mark indicates that no time-step cuts were observed. A maximum time-step size of 5 days was used for Case 3 where time-step cuts were observed for the approximate approach. Finally, Case 4 uses a time-step size of 10 days with time-step cuts observed for both exact and approximate approaches.

Table 1: Comparison of total and linear solver CPU times for Exact and Approximate approaches

	Exact		Approximate	
	Total time	Solver time	Total time	Solver time
Case 1	307.2	288.2	74.6	47.3
Case 2	137.6	132.2	34.6	25.1
Case 3	87.3	85.3	40.6	34.4
Case 4	79.1	77.3	28.5	23.6

A computational speedup of approximately 4 times was observed for cases 1 thru 4; as shown in table 1, and is primarily due to a reduction in computational overheads for the linear solver. Figure 3 shows the saturation profile after 500 days of continuous water injection for the exact (left) and approximate (right) approaches. Further, figures 4 and 5 show the spatial distributions of the absolute value of the difference in saturation profiles (error norm); after 500 days, obtained from the two approaches for Case 1 thru 4.

Table 2: Nonlinear solver convergence for Exact and Approximate approaches with varying time-step sizes

	α	β	Δt_{min}	Δt_{max}	Exact	Approximate
Case 1	1.0	0.5	0.1	0.2	✓	✓
Case 2	1.01	0.5	0.1	1.0	✓	✓
Case 3	1.05	0.5	0.1	5.0	✓	-
Case 4	1.05	0.5	0.1	10.0	-	-

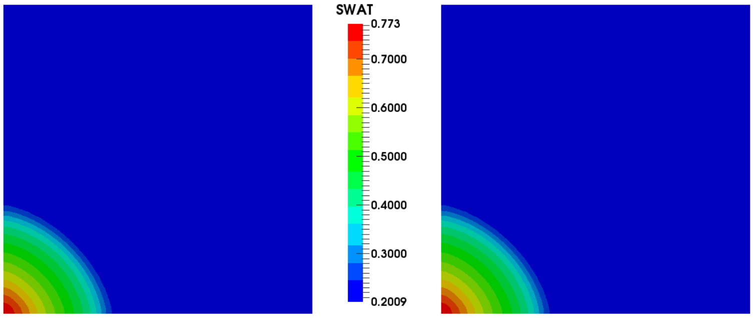

Figure 3: Matching saturation profiles at the end of 500 days for exact (left) and approximate (right) Jacobian approaches.

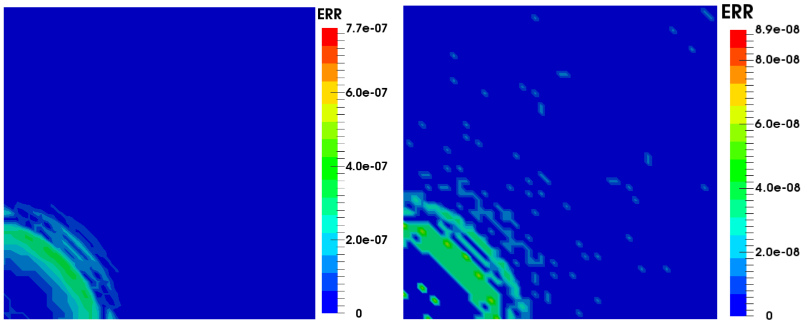

Figure 4: Absolute of saturation difference for test Cases 1 (left) and 2 (right) at the end of 500 days.

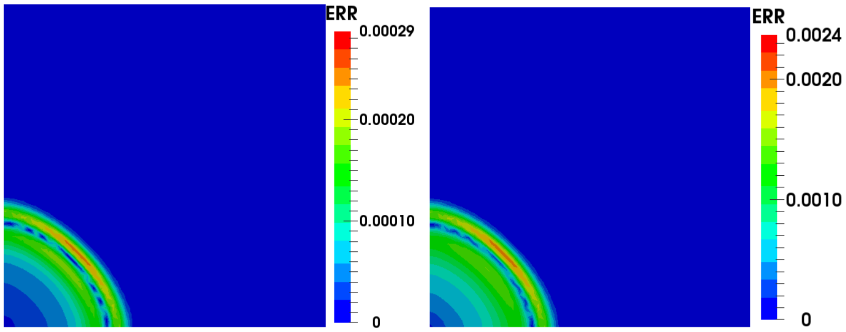

Figure 5: Absolute of saturation difference for test Cases 3 (left) and 4 (right) at the end of 500 days.

Please note that the differences in the absolute values of the saturation distributions for Cases 3 and 4 are expected due to differences in time-step cuts between exact and approximate approaches. These differences arise due to unequal numerical diffusion introduced by different time-step sizes. In fact, as we go from Case 1 to 4 in Figures 4 and 5 the saturation difference between the two approaches increases. However, for each of these approaches the nonlinear residuals converge to the same relative, nonlinear tolerance of $1{\times}10^{-}8$. Since large time-step sizes also introduce large numerical diffusions, in what follows we have selected appropriate maximum time-step sizes that allow us to compare the two approaches while avoiding unreasonably diffuse solutions. We do not report the production oil rates, water cuts and cumulative recoveries as these aforementioned differences are almost negligible and cannot be identified using these plots.

8.2 Kueper Sandbox Problem

Next, we use a modified Kueper sandbox problem with strong counter-current flow due to capillary pressure and density differences. The setup contains four different rock types with different capillary pressure and relative permeability curves defined using Brooks-

Corey model. The computational domain is 50cm×70cm×1cm uniformly discretized using 50×70×1 elements. In this numerical experiment, we use the computational domain along with rock and reservoir properties as described in the original work by [17]. Figure 6 shows a schematic of the Kueper sandbox problem with four different rock types (or sand-packs) with different relative permeability and capillary pressure curves.

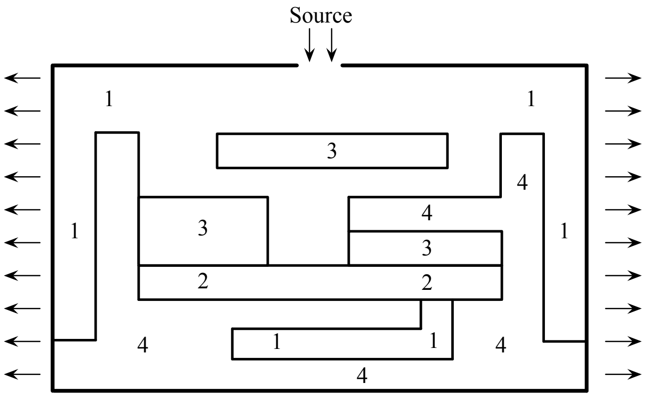

Figure 6: Rock type distribution for the Kueper sandbox problem.

The initial conditions for pressure and water saturation are taken to be 0.2 and 100 psi, respectively. We assume a no-flow boundary condition on the entire domain. A pressure specified injection well is place at the top-middle of the computational domain with 5 equidistant pressure specified production (or boundary) wells each at the left and right boundaries of the domain to mimic the original Kueper sandbox problem. The injection and production well pressure specification is set at 100.1 psi and 100 psi, respectively. The pressure difference between the source and sink is purposely kept small (at 0.1 psi) so that the flow is mostly gravity and capillary pressure dominated.

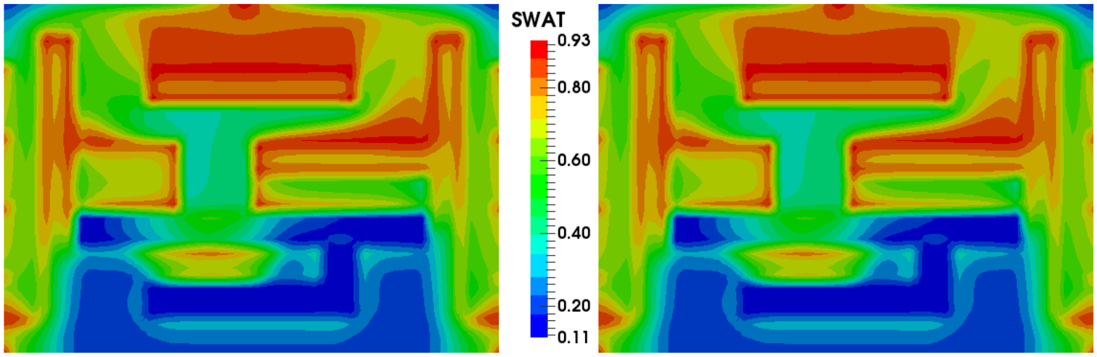

Figure 7: Matching saturation profiles at the end of 0.005 days for exact (left) and approximate Jacobian approaches

The numerical simulations are ran for a total duration of 0.005 days (or 7.2 minutes) for both exact and approximate Jacobian approaches. The initial, minimum, and maximum time-step sizes for the two approaches are kept the same and are taken to be 10^{-8}, 10^{-8}, and 10^{-5}, respectively. Further, the time-step augmentation and reduction factors are taken to be 0.5 and 1.01, respectively. We observed nonlinear iteration failures; leading to time-step cuts, occurring at different times for both the approaches. Figure 7 shows the saturation profiles at the end of 0.005 days for the exact (left) and approximate (right) Jacobian approaches. Further, Figure 8 shows the absolute difference in the saturation profiles at the same time. The total and linear solver times are 1300.755 and 1207.336, respectively for the exact approach, and 604.923 and 508.230, respectively for the approximate approach. As can be seen, the speedup obtained by approximate Jacobian approach is approximately 2.15 times and is primarily due to reduced computational overheads for the linear solver. These numerical results demonstrate that the solution accuracy of the approximate against the exact approach.

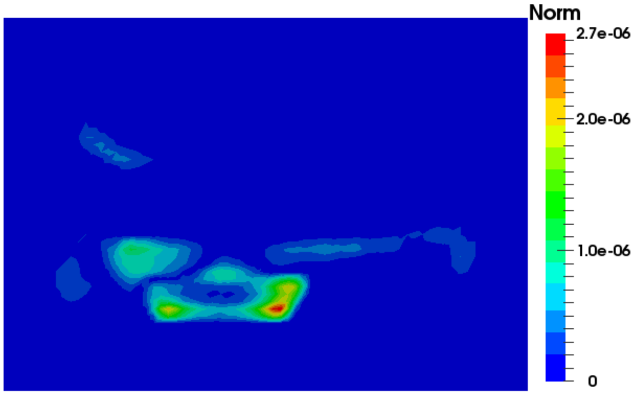

Figure 8: Absolute of saturation difference at the end of 0.005 days between exact and approximate Jacobian approaches

8.3 Brugge Field Water Flooding

We now consider a field scale water flooding scenario with reservoir geometry, rock properties, and well placement taken from the Brugge field history matching study [8]. The rock properties are heterogeneous and include distributions of permeability and porosity as well as different rock types with varying relative permeability and capillary pressure descriptions. The numerical reservoir model consists of 7172 grid elements with no-flow external boundaries. The reader is referred to [8] for further details regarding reservoir property description. The fluid densities are taken to be 56.0 and 62.6 lb/ft^3 with viscosities 1.0 and 1.29 cP for the oil and water phases, respectively. Further, the water and oil phase compressibilities are 3×10^{-6} and 9.26×10^{-6} psi^{-1}, respectively. The initial conditions for pressure and saturation are evaluated using an equilibrium calculations consistent with the estimated original oil in place.

The Brugge field scenario consists of 10 injection and 20 production wells with bottom-hole pressure specification of 4000 psi and 2000 psi, respectively. These modified specifications are chosen to prevent well-shut down during the simulation run for both exact and approximate Jacobian approaches. Additionally, the gravity vector is taken to be in the positive x direction as shown in Figure 9. The numerical simulations are performed for a total of 7301 days or approximately 20 years with initial, minimum, and maximum time-step sizes of 0.1, 0.1, and 10 days, respectively. The time-step size augmentation and reduction factors are further taken to be 1.1 and 0.5, respectively. The total and linear solver CPU times are observed to be 98.481 and 88.234, and 74.044 and 50.577, respectively for the exact and approximate Jacobian approaches. The approximate approach is therefore 1.32 times faster than the exact Jacobian approach.

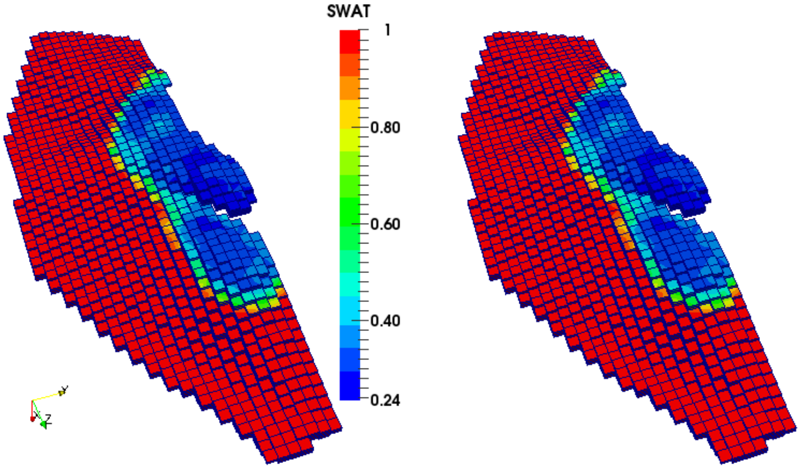

Figure 9: Matching saturation profiles at the end of 7301 days for exact (left) and approximate Jacobian approaches

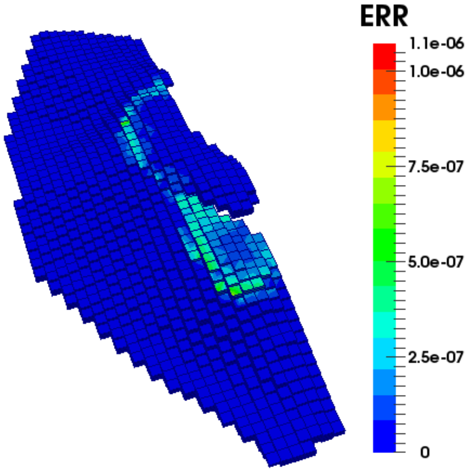

Figure 10: Absolute of saturation difference at the end of 7301 days between exact and approximate Jacobian approaches

Figure 9 shows the saturation profiles after 7301 days obtained using exact (left) and approximate (right) Jacobian approaches. The absolute differences in saturation distributions at the end of the simulation is also show in Figure 10. As mentioned before, although the nonlinear solver tolerances for the two approaches are the same, the differences occur due to differences in time-step cuts and augmentations during the simulation run. The order of magnitude of these differences remain the same during the simulation run.

8.4 Stuttgart Field Water Flooding

Next, we consider another field scale water flooding case with reservoir geometry obtained using the Stuttgart field. The computational domain in this problem models the Johansen formation off the coast of Norway and is a stair-stepped approximation of the original field. It is discretized into 12166 grid elements with no flow boundary conditions as before. The heterogeneous porosity and permeability distributions are obtained from the original dataset which can be downloaded at [1]. The capillary pressure is assumed to be identically zero with oil and water phase relative permeability variation with water saturation shown in Figure 11. A single pressure specified injection well is considered in the centre of the reservoir domain with a bottom-hole pressure specification of 7000 psi. Further, 34 production wells are placed equidistantly at the external reservoir boundaries with bottom-hole pressure specifications ranging from 3953 to 4841 psi. The fluid densities are 62.6 and 56. lb/ft^3 with viscosities 0.5 and 2.0 cP and compressibilities 1×10$^-$6 and 1×10$^-$4 psi$^-$1 for the water and oil phases, respectively. The initial conditions are specified at 7000 psi and 0.2 for the initial pressure and water saturation, respectively.

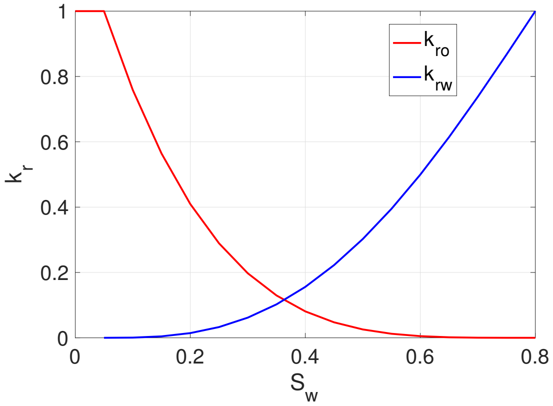

Figure 11: Oil and water phase relative permeability curves for the Stuttgart field case.

The numerical simulation is performed for a total duration of 4000 days; approximately 11 years, with initial, minimum, and maximum time-step sizes of 0.2, 0.2 and 5.0, respectively. Again the time-step size reduction and augmentation factors are kept the same as before at 0.5 and 1.1, respectively. The total and linear solver CPU times are observed to be 206.755 and 188.465, and 59.757 and 42.836, respectively for the exact and approximate Jacobian approaches. A computational speedup of 3.5 times is obtained for the approximate approach and can be easily observed from the computational savings in the linear solver time. Figure 12 shows the water saturation distributions; after 200 days, using exact (left) and approximate (right) Jacobian appraoches. The absolute differences in the two saturation distributions is also shown in Figure 13.

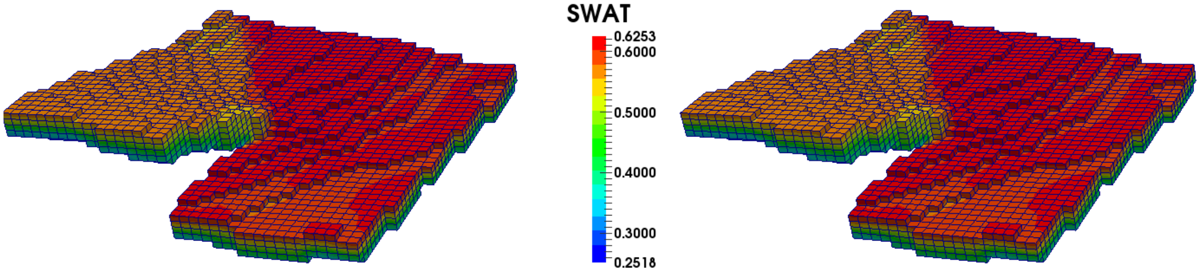

Figure 12: Matching saturation profiles at the end of 200 days for exact (left) and approximate Jacobian approaches

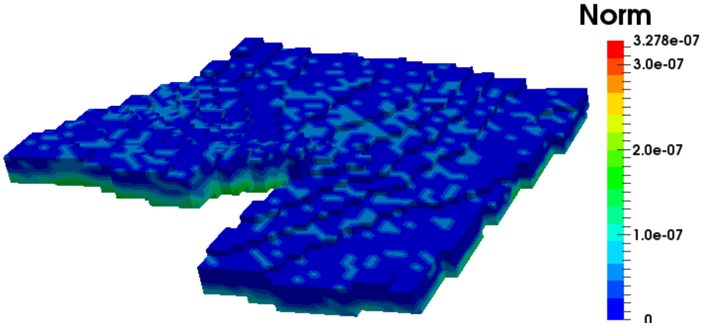

Figure 13: Absolute of saturation difference at the end of 200 days between exact and approximate Jacobian approaches

9 Conclusions

We presented an approximate Jacobian construction as an alternative to the conventional Newton method; with exact Jacobian construction, as a nonlinear solver. The proposed approach is a fully implicit, nonlinear solver for coupled multiphase flow and transport problems that is easier to implement compared to the conventional method since it does not require construction of tedious contributions to the Jacobian matrix. The approximation is made in an effort to diagonalize a few sub-matrices in the monolithic system allowing trivial elimination (Schur complement) of the saturation unknowns. The resulting reduced system in the pressure unknown is a positive definite matrix that can be solved using conventional linear solvers such as GMRES with AMG preconditioner. Our numerical results indicated computational speedups ranging from 1.32 to 4 times when the approximate Jacobian construction is used as opposed to the exact Jacobian construction. The results consistently indicate a large computational cost reduction in the linear solver resulting in overall gain in efficiency. The exact Jacobian approach converges for larger time-steps and requires lesser nonlinear iterations for a given time-step size when compared to our approach. However, the overall computational gain in the linear solver times with approximate Jacobian construction allows it to surpass the conventional method during our extensive numerical benchmarking. We demonstrate computational speedups for a wide range of numerical experiments considering heterogeneous rock properties such as permeability, porosity, capillary pressure, and relatively permeability. The Kueper sandbox problem demonstrates the capability of the approximate Jacobian approach in solving multiphase, counter-current flow problems dominated by capillary pressure and buoyancy forces.

Acknowledgements

The first and last authors were funded by the DOE-NETL grant DE-FE0023314. The second author is funded by NSF grant 1546553 and KAUST excellence fund. We are grateful for the opportunity and the support provided for this endeavor. We would also like to thank the Center for Subsurface Modeling industrial affiliates for their continuing support.

A Appendix

We present here details on the quadrature rules discussed in Section 3 and their use to reduce equations (25) to a 5-point cell-centered pressure system. The derivation is shown only for the case $d = 2$; the generalization to $d = 3$ is straightforward. We use relatively standard cell-centered finite difference notation. Let us consider a rectangular grid with grid points denoted by

$$(x_{i+1/2}, y_{j+1/2}), \quad i = 0, \cdots, N_x, j = 0, \cdots, N_y,$$

and then define

$$\begin{aligned} h_i^x &= x_{i+1/2} - x_{i-1/2}, \quad i = 1, \cdots, N_x, \\ h_j^y &= y_{j+1/2} - y_{j-1/2}, \quad j = 1, \cdots, N_y. \end{aligned}$$

An element is denoted by $E_{i,j}$ with $|E_{i,j}|$ being its measure (area for $d = 2$ or volume for $d = 3$). An edge of the grid is represented by $e_{i+1/2,j}$ (or $e_{i,j+1/2}$) and $|e_{i+1/2,j}|$ stands for its length/area. With each edge $e_{i+1/2,j}$ we associate one basis vector from the RTN_0 space, with one of the components a“hat” function and the other one zero. We denote this function by $\varphi_{i+1/2,j}$ (dotted blue in Figure 14) and define it by

$$\varphi_{i+1/2,j} = \begin{cases} \dfrac{x - x_{i-1/2}}{|E_{i,j}|}, & \text{on } E_{i,j}, \\ \dfrac{x_{i+3/2} - x}{|E_{i+1,j}|}, & \text{on } E_{i+1,j}. \end{cases} \tag{33}$$

Note that, by construction

$$\varphi_{m+1/2,j}(e_{i+1/2,j}) = \begin{cases} \dfrac{1}{|e_{i+1/2,j}|}, & m = i, \\ 0, & m \neq i. \end{cases} \tag{34}$$

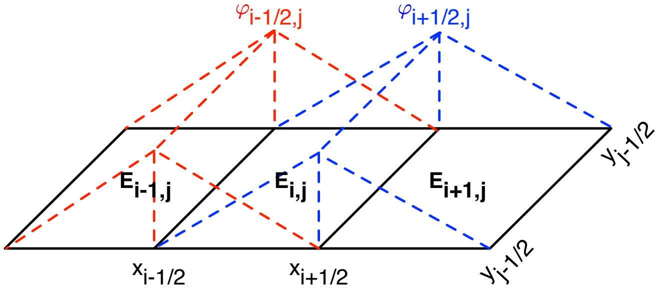

Figure 14: "Hat" functions associated with edges $e_{i-1/2,j}$ and $e_{i+1/2,j}$

In y-direction, functions $\{\varphi_{i,j+1/2}\}$ are defined in a similar fashion. For basis of the pressure component of the RTN$_0$ space, we make the choice

$$w_{i,j} = \begin{cases} 1, & \text{on } E_{i,j}, \\ 0, & \text{otherwise.} \end{cases}$$

We approximate

$$(K^{-1}\boldsymbol{u}_h, \boldsymbol{v}_h) \approx (K^{-1}\boldsymbol{u}_h, \boldsymbol{v}_h)_{\text{TM}}. \tag{35}$$

where, as in [3], $(\cdot,\cdot)_M$ and $(\cdot,\cdot)_T$ represents the mid-point and trapezoidal rules of integration, respectively (in each of the co-ordinate direction) and for $\mathbf{v}, \mathbf{q} \in \mathbb{R}^d$,

$$(\mathbf{v}, \mathbf{q})_{\text{TM}} = \left\{ \begin{array}{ll} (v_1, q_1)_{T\times M} + (v_2, q_2)_{M\times T} & \text{if } d = 2, \\ (v_1, q_1)_{T\times M\times M} + (v_2, q_2)_{M\times T\times M} + (v_3, q_3)_{M\times M\times T} & \text{if } d = 3. \end{array} \right. \tag{36}$$

In order to evaluate the integrals in Eqn. (25), we first calculate $(\varphi_{i\pm 1/2,j}, \varphi_{i+1/2,j})_{\text{TM}}$ using the quadrature rules (36):

$$\begin{aligned} (\varphi_{i+1/2,j}, \varphi_{i+1/2,j})_{\text{TM}} &= \frac{1}{2}|E_{i,j}|\left(0^2 + \frac{1}{|e_{i+1/2,j}|^2}\right) + \frac{1}{2}|E_{i+1,j}|\left(\frac{1}{|e_{i+1/2,j}|^2} + 0^2\right) \\ &= \frac{h_i^x}{2\,|e_{i+1/2,j}|} + \frac{h_{i+1}^x}{2\,|e_{i+1/2,j}|} = \frac{h_i^x + h_{i+1}^x}{2\,|e_{i+1/2,j}|}, \end{aligned} \tag{37}$$

$$(\varphi_{i-1/2,j}, \varphi_{i+1/2,j})_{\text{TM}} = \frac{1}{2}|E_{i,j}|\left(\frac{1}{|e_{i-1/2,j}|}.0 + 0.\frac{1}{|e_{i+1/2,j}|}\right) = 0. \tag{38}$$

Note that using a similar argument, we get $(\varphi_{i+3/2,j}, \varphi_{i+1/2,j})_{\mathrm{TM}} = 0$. The solution of Eqn. (25) can be written in $2d$ as $\boldsymbol{u}_\alpha = [u_\alpha^x\,, u_\alpha^y]^T$, $\tilde{\boldsymbol{u}}_\alpha = [\tilde{u}_\alpha^x\,, \tilde{u}_\alpha^y]^T$, p_o, and S_α with $\alpha = o, w$, where

$$u_\alpha^x = \sum_{m=0}^{N_x} \sum_{n=1}^{N_y} U^\alpha_{m+1/2,n}\, \varphi_{m+1/2,n} \qquad u_\alpha^y = \sum_{m=1}^{N_x} \sum_{n=0}^{N_y} U^\alpha_{m,n+1/2}\, \varphi_{m,n+1/2} \tag{39a}$$

$$\tilde{u}_\alpha^x = \sum_{m=0}^{N_x} \sum_{n=1}^{N_y} \tilde{U}^\alpha_{m+1/2,n}\, \varphi_{m+1/2,n} \qquad \tilde{u}_\alpha^y = \sum_{m=1}^{N_x} \sum_{n=0}^{N_y} \tilde{U}^\alpha_{m,n+1/2}\, \varphi_{m,n+1/2} \tag{39b}$$

$$p_o = \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} P_{m,n} w_{m,n} \qquad S_\alpha = \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} S^\alpha_{m,n} w_{m,n}. \tag{39c}$$

First, we plug $w = w_{i,j}$ in Eqn. (25a). The first term trivially becomes

$$\begin{aligned}(\phi\rho_\alpha S_\alpha, w_{i,j}) &= \left(\phi\rho_\alpha(p_\alpha) \sum_{n=1}^{N_y} S^\alpha_{m,n} w_{m,n}), w_{i,j} \right) \\ &= \int_{E_{i,j}} \phi\rho_\alpha(p_\alpha)\, (S^\alpha_{i,j}\, w_{i,j})\, w_{i,j} = \phi\rho_\alpha(p_\alpha) S^\alpha_{i,j}\, |E_{i,j}|. \end{aligned} \tag{40}$$

After applying divergence theorem and using (39) and (34), we get

$$\begin{aligned}(\nabla\cdot \boldsymbol{u}_\alpha, w_{i,j}) &= \int_{E_{i,j}} \nabla\cdot\boldsymbol{u}_\alpha = \int_{\partial E_{i,j}} \boldsymbol{u}_\alpha \cdot \boldsymbol{n}_{E_{i,j}} \\ &= \int_{e_{i+1/2,j}} u_\alpha^x - \int_{e_{i-1/2,j}} u_\alpha^x + \int_{e_{i,j+1/2}} u_\alpha^y - \int_{e_{i,j-1/2}} u_\alpha^y \\ &= \int_{e_{i+1/2,j}} \frac{U^\alpha_{i+1/2,j}}{|e_{i+1/2,j}|} - \int_{e_{i-1/2,j}} \frac{U^\alpha_{i-1/2,j}}{|e_{i-1/2,j}|} + \int_{e_{i,j+1/2}} \frac{U^\alpha_{i,j+1/2}}{|e_{i,j+1/2}|} - \int_{e_{i,j-1/2}} \frac{U^\alpha_{i,j-1/2}}{|e_{i,j-1/2}|} \\ &= U^\alpha_{i+1/2,j} - U^\alpha_{i-1/2,j} + U^\alpha_{i,j+1/2} - U^\alpha_{i,j-1/2}. \end{aligned} \tag{41}$$

If we define $q^\alpha_{i,j} = \displaystyle\int_{E_{i,j}} q_\alpha$, then Eqn. (25a) reads:

$$\frac{\phi\rho_\alpha(p_\alpha) S^\alpha_{i,j} - \Big(\rho_\alpha(p_\alpha)\, S^\alpha_{i,j}\Big)^n}{\Delta t} |E_{i,j}| + U^\alpha_{i+1/2,j} - U^\alpha_{i-1/2,j} + U^\alpha_{i,j+1/2} - U^\alpha_{i,j-1/2} = q^\alpha_{i,j}. \tag{42}$$

We next denote by $K_{i,j} = \begin{bmatrix} k^x_{i,j} & 0 \\ 0 & k^y_{i,j} \end{bmatrix}$ the rock permeability on element $E_{i,j}$ and we plug a basis vector, say $\boldsymbol{v} = \big[\varphi_{i+1/2,j}\,, 0\big]^T$, in Eqn. (25c) and using the quadrature rules (36) we get:

$$\begin{aligned}(K^{-1}\tilde{\boldsymbol{u}}_w, \boldsymbol{v}) \approx (K^{-1}\tilde{\boldsymbol{u}}_w, \boldsymbol{v})_{\mathrm{TM}} &= ((k^x)^{-1} \tilde{\boldsymbol{u}}^x_w, \varphi_{i+1/2,j})_{\mathrm{TM}} \\ &= \sum_{m=i-1}^{i+1} \tilde{U}^w_{m+1/2,j} ((k^x)^{-1} \varphi_{m+1/2,j}, \varphi_{i+1/2,j})_{\mathrm{TM}}, \end{aligned} \tag{43}$$

where the only non-zero terms in the above sum, except at the external boundaries, are the ones for which $\varphi_{m+1/2,n}$ is non-zero on $E_{i,j}$ or $E_{i+1,j}$. Following (37) and (38), we obtain

$$\begin{aligned}
((k^x)^{-1}\varphi_{i+1/2,j},\varphi_{i+1/2,j})_{\mathrm{TM}} &= \frac{1}{2}|E_{i,j}|\left(0^2+\frac{(k^x_{i,j})^{-1}}{|e_{i+1/2,j}|^2}\right)+\frac{1}{2}|E_{i+1,j}|\left(\frac{(k^x_{i+1,j})^{-1}}{|e_{i+1/2,j}|^2}+0^2\right)\\
&= \frac{h^x_i}{2\,|e_{i+1/2,j}|k^x_{i,j}}+\frac{h^x_{i+1}}{2\,|e_{i+1/2,j}|k^x_{i+1,j}} \qquad (44)\\
&= \frac{1}{2\,|e_{i+1/2,j}|}\left(\frac{h^x_i}{k^x_{i,j}}+\frac{h^x_{i+1}}{k^x_{i+1,j}}\right),
\end{aligned}$$

$$((k^x)^{-1}\varphi_{i-1/2,j},\varphi_{i+1/2,j})_{\mathrm{TM}} = \frac{1}{2}|E_{i,j}|\left(\frac{(k^x_{i,j})^{-1}}{|e_{i-1/2,j}|}.0+0.\frac{1}{|e_{i+1/2,j}|}\right)=0. \qquad (45)$$

Putting together (43)-(45) results in

$$(K^{-1}\tilde{\boldsymbol{u}}_w,\boldsymbol{v})_{\mathrm{TM}}=\frac{1}{2\,|e_{i+1/2,j}|}\left(\frac{h^x_i}{k^x_{i,j}}+\frac{h^x_{i+1}}{k^x_{i+1,j}}\right)\tilde{U}^w_{i+1/2,j}, \qquad (46)$$

i.e., after Newton linearization, the approximate matrix A in system (28) has only one non-zero entry in the row corresponding to the velocity unknown $\delta\tilde{U}^w_{i+1/2,j}$, resulting in a diagonal matrix. Please note that the entries of this matrix are weighted harmonic average of the permeabilities adjacent to an element edge, as can be seen in Eqn. (46). It is trivial to see that,

$$\frac{\partial}{\partial x}\varphi_{i+1/2,j}=\begin{cases}\dfrac{1}{|E_{i,j}|}, & \text{on } E_{i,j},\\[2ex] \dfrac{-1}{|E_{i+1,j}|}, & \text{on } E_{i+1,j},\end{cases}$$

so the second term in Eqn. (25c) becomes

$$(p_o,\nabla\cdot\boldsymbol{v})=\left(P_{i,j},\frac{1}{|E_{i,j}|}\right)_{E_{i,j}}+\left(P_{i+1,j},\frac{-1}{|E_{i+1,j}|}\right)_{E_{i+1,j}}=P_{i,j}-P_{i+1,j}. \qquad (47)$$

That means that the matrix B in system (28) has (at most) two non-zero entries, 1 and -1, in the row corresponding to the velocity through face $e_{i+1/2,j}$, at the columns corresponding to pressure at the two adjacent elements $E_{i,j}$ and $E_{i+1,j}$. Using the same argument, the capillary pressure term in Eqn. (25c) can be written as:

$$(p_c,\nabla\cdot\boldsymbol{v})=\left(p_c(S_w),\frac{1}{|E_{i,j}|}\right)_{E_{i,j}}+\left(p_c(S_w),\frac{-1}{|E_{i+1,j}|}\right)_{E_{i+1,j}}=p_c(S^w_{i,j})-p_c(S^w_{i+1,j}). \qquad (48)$$

Next, to express the term due to gravity in Eqn. (25c), we write $\boldsymbol{g} = [g^x\,, g^y]^T$, so:

$$\begin{aligned}(\rho_w \boldsymbol{g}, \boldsymbol{v}) \approx (\rho_w \boldsymbol{g}, \boldsymbol{v})_{\mathrm{TM}} &= (\rho_w g^x, \varphi_{i+1/2,j})_{\mathrm{TM}} \\ &= \frac{g^x}{2}|E_{i,j}| \left(0 + \frac{\rho_w(P_{i,j} - p_c(S^w_{i,j}))}{|e_{i+1/2,j}|^2}\right) + \frac{1}{2}|E_{i+1,j}| \left(\frac{\rho_w(P_{i+1,j} - p_c(S^w_{i+1,j}))}{|e_{i+1/2,j}|^2} + 0\right) \\ &= \frac{1}{2} g^x \Big[h^x_i\, \rho_w\big(P_{i,j} - p_c(S^w_{i,j})\big) + h^x_{i+1}\, \rho_w\big(P_{i+1,j} - p_c(S^w_{i+1,j})\big)\Big]. \end{aligned} \tag{49}$$

Putting together (46)-(49), the water phase mass balance equation (25c) with $\boldsymbol{v} = \big[\varphi_{i+1/2,j}\,, 0\big]^T$ reads:

$$\begin{aligned}\frac{1}{2\,|e_{i+1/2,j}|}\left(\frac{h^x_i}{k^x_{i,j}} + \frac{h^x_{i+1}}{k^x_{i+1,j}}\right)\tilde{U}^w_{i+1/2,j} - P_{i,j} + P_{i+1,j} &= -p_c(S^w_{i,j}) + p_c(S^w_{i+1,j}) \\ &+ \frac{1}{2} g^x \Big[h^x_i\, \rho_w\big(P_{i,j} - p_c(S^w_{i,j})\big) + h^x_{i+1}\, \rho_w\big(P_{i+1,j} - p_c(S^w_{i+1,j})\big)\Big]. \end{aligned} \tag{50}$$

Following the same arguments, the oil phase mass balance equation (25b) with the same basis vector $\boldsymbol{v}$ is now

$$\frac{1}{2\,|e_{i+1/2,j}|}\left(\frac{h^x_i}{k^x_{i,j}} + \frac{h^x_{i+1}}{k^x_{i+1,j}}\right)\tilde{U}^o_{i+1/2,j} - P_{i,j} + P_{i+1,j} = \frac{1}{2} g^x \left[h^x_i\, \rho_o\left(P_{i,j}\right) + h^x_{i+1}\, \rho_o\left(P_{i+1,j}\right)\right]. \tag{51}$$

We now turn our attention to the auxiliary in system (25). Plugging $\boldsymbol{v} = \big[\varphi_{i+1/2,j}\,, 0\big]^T$ in the left-hand side of Eqn. (25d), we get:

$$\begin{aligned}(\boldsymbol{u}_\alpha, \boldsymbol{v}) \approx (\boldsymbol{u}_\alpha, \boldsymbol{v})_{\mathrm{TM}} &= (u^x_\alpha, \varphi_{i+1/2,j})_{\mathrm{TM}} \\ &= \sum_{m=i-1}^{i+1} U^\alpha_{m+1/2,j} (\varphi_{m+1/2,j}, \varphi_{i+1/2,j})_{\mathrm{TM}} = \frac{h^x_i + h^x_{i+1}}{2\,|e_{i+1/2,j}|}\, U^\alpha_{i+1/2,j}. \end{aligned} \tag{52}$$

Lastly, for the second term in Eqn. (25d) we use the standard choice of upwind mobility in combination with the trapezoidal-midpoint rule (36) to obtain:

$$(\lambda_\alpha \tilde{\boldsymbol{u}}_\alpha, \boldsymbol{v}) \approx (\lambda^*_\alpha \tilde{\boldsymbol{u}}, \boldsymbol{v})_{\mathrm{TM}} = \frac{h^x_i + h^x_{i+1}}{2\,|e_{i+1/2,j}|}\, \lambda^{\alpha,*}_{i+1/2,j} \tilde{U}^\alpha_{i+1/2,j}, \tag{53}$$

which results in the simplified form

$$U^\alpha_{i+1/2,j} = \lambda^{\alpha,*}_{i+1/2,j} \tilde{U}^\alpha_{i+1/2,j}. \tag{54}$$

Here, the upwind mobility is defined as

$$\lambda^{\alpha,*}_{i+1/2,j} = \rho^{\alpha,*}_{i+1/2,j} \frac{k^{r\alpha,*}_{i+1/2,j}}{\mu_\alpha} = \begin{cases} \dfrac{1}{2\mu_\alpha}\left(\rho^\alpha_{i,j} + \rho^\alpha_{i+1,j}\right) k_{r\alpha}(S^\alpha_{i,j}), & \text{if } \ \tilde{U}^\alpha_{i+1/2,j} > 0, \\ \dfrac{1}{2\mu_\alpha}\left(\rho^\alpha_{i,j} + \rho^\alpha_{i+1,j}\right) k_{r\alpha}(S^\alpha_{i+1,j}), & \text{otherwise}, \end{cases} \tag{55}$$

where

$$\rho^o_{i,j} = \rho_o(P_{i,j}), \quad \rho^w_{i,j} = \rho_w\big(P_{i,j} - p_c(S^w_{i,j})\big).$$

B Appendix

Let us consider an operator of the form

$$\mathcal{F}(p) = \nabla\cdot\big(\rho(p)\, K\nabla p\big),$$

where $\rho = \rho(p)$ is given. Using (9), the Gâteaux derivative of $\mathcal{F}(p)$ at p and in the direction δp is:

$$\begin{aligned}
\mathcal{F}'(p)\delta p &= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\Big\{\nabla\cdot\Big(\rho(p+\varepsilon\delta p)\, K\nabla(p+\varepsilon\delta p)\Big) - \nabla\cdot\big(\rho(p)\, K\nabla p\big)\Big\} \\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\nabla\cdot\Big\{\big[\rho(p) + \rho'(p)\varepsilon\delta p + O(\varepsilon^2)\big]\,\big[K\nabla p + \varepsilon K\nabla\delta p\big] - \rho(p)\, K\nabla p\Big\} \\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\nabla\cdot\Big\{\varepsilon\rho'(p)\delta p\, K\nabla p + \varepsilon\rho(p)\, K\nabla\delta p + O(\varepsilon^2)\Big\} \\
&= \nabla\cdot\big(\rho'(p)\delta p\, K\nabla p + \rho(p)\, K\nabla\delta p\big)
\end{aligned}$$

Here we used a Taylor expansion for the function ρ as well as the linearity of the operators ∇, $\nabla\cdot$, and multiplication by K. Using the short-hand notation of the δ operator, we can rewrite this as

$$\delta\big(\nabla\cdot\big(\rho(p)\, K\nabla p\big)\big) = \nabla\cdot\big(\rho'(p)\delta p\, K\nabla p + \rho(p)\, K\nabla\delta p\big),$$

that resembles the derivative of a real-valued function.

References

[1] Workshop on numerical models for CO_2 storage in geological formations. http://www.iws.uni-stuttgart.de/co2-workshop.

[2] Todd Arbogast, Philip T. Keenan, Mary F. Wheeler, and Ivan Yotov. Logically rectangular mixed methods for Darcy flow on general geometry. *SPE Journal*, SPE-29099-MS:51–59, 1982.

[3] Todd Arbogast, Mary F. Wheeler, and Ivan Yotov. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. *SIAM Journal on Numerical Analysis*, 34(2):828–852, 1997.

[4] Randolph E. Bank and Donald J. Rose. Global approximate Newton methods. *Numerische Mathematik*, 37(2):279–295, 1981.

[5] Randolph E. Bank and Christian Wagner. Multilevel ILU decomposition. *Numerische Mathematik*, 82(4):543–576, 1999.

[6] Paul T. Boggs, Jon W. Tolle, and Pyng Wang. On the local convergence of quasi-Newton methods for constrained optimization. *SIAM Journal on Control and Optimization*, 20(2):161–171, 1982.

[7] Hui Cao, Hamdi A. Tchelepi, John Richard Wallis, and Hrant E. Yardumian. Parallel scalable unstructured CPR-type linear solver for reservoir simulation. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2005.

[8] Yan Chen and Dean S. Oliver. Ensemble-based closed-loop optimization applied to Brugge field. *SPE Reservoir Evaluation & Engineering*, 13(01):56–71, 2010.

[9] Clint N. Dawson, Héctor Klíe, Mary F. Wheeler, and Carol S. Woodward. A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton–Krylov solver. *Computational Geosciences*, 1(3):215–249, 1997.

[10] Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

[11] John E. Dennis Jr. and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[12] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

[13] Harold Carter Edwards. A parallel multilevel-preconditioned GMRES solver for multiphase flow models in the Implicit Parallel Accurate Reservoir Simulator (IPARS). Technical Report TICAM 98-04, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1998.

[14] François P. Hamon and Hamdi A. Tchelepi. Ordering-based nonlinear solver for fully-implicit simulation of three-phase flow. *Computational Geosciences*, 20(3):475–493, 2016.

[15] Carl T. Kelley. *Iterative methods for linear and nonlinear equations*. Society for Industrial and Applied Mathematics, 1995. pp. 76–78.

[16] Héctor Klíe, Marcelo Rame, and Mary F. Wheeler. Two-stage preconditions for Inexact Newton methods in multi-phase reservoir simulation. Technical Report CRPC-TR96660, Center for Research on Parallel Computation, Rice University, 1996.

[17] Bernard H. Kueper and Emil O. Frind. Two-phase flow in heterogeneous porous media: 1. Model development. *Water Resources Research*, 27(6):1049–1057, 1991.

[18] Felix Kwok and Hamdi A. Tchelepi. Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media. *Journal of Computational Physics*, 227(1):706–727, 2007.

[19] Sébastien Lacroix, Yuri V. Vassilevski, John Wheeler, and Mary F. Wheeler. Iterative solution methods for modeling multiphase flow in porous media fully implicitly. *SIAM Journal on Scientific Computing*, 25(3):905–926, 2003.

[20] Sébastien Lacroix, Yuri V. Vassilevski, and Mary F. Wheeler. Decoupling preconditioners in the Implicit Parallel Accurate Reservoir Simulator (IPARS). *Numerical Linear Algebra with Applications*, 8(8):537–549, 2001.

[21] Leonid P. Lebedev and Iosif I. Vorovich. Elements of nonlinear functional analysis. In *Functional Analysis in Mechanics*, pages 177–218. Springer New York, 2003.

[22] Mie Nakata, Alan Weiser, and Mary F. Wheeler. Some superconvergence results for mixed finite element methods for elliptic problems on rectangular domains. *The Mathematics of Finite Elements and Applications V*, MAFELAP:51–59, 1985.

[23] J.-C. Nédélec. Mixed finite elements in R^3. *Numerische Mathematik*, 35(3):315–341, 1980.

[24] Małgorzata Peszyńska, Mary F. Wheeler, and Ivan Yotov. Mortar upscaling for multiphase flow in porous media. *Computational Geosciences*, 6(1):73–100, 2002.

[25] Liqun Qi and Jie Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(1):353–367, 1993.

[26] Pierre-Arnaud Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics*, volume 606, pages 292–315. Springer-Verlag, New York, 1977.

[27] John W Ruge and Klaus Stüben. Algebraic multigrid. *Multigrid Methods*, 3(13):73–130, 1987.

[28] Thomas F. Russell and Mary F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. *The Mathematics of Reservoir Simulation*, 1:35–106, 1983.

[29] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[30] Gurpreet Singh and Mary F. Wheeler. Compositional flow modeling using a multipoint flux mixed finite element method. *Computational Geosciences*, 20(3):421–435, 2016.

[31] Klaus Stüben. An introduction to algebraic multigrid. *Multigrid*, pages 413–532, 2001.

[32] John Richard Wallis, R. P. Kendall, and T. E. Little. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1985.

[33] Mary F. Wheeler and Ivan Yotov. A multipoint flux mixed finite element method. *SIAM Journal on Numerical Analysis*, 44(5):2082–2106, 2006.

[34] Rami Younis, Hamdi A. Tchelepi, and Khalid Aziz. Adaptively localized continuation-Newton method–nonlinear solvers that converge all the time. *SPE Journal*, 15(2), 2010.