Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 706–727
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media

Felix Kwok, Hamdi Tchelepi *

Stanford University, Energy Resources Engineering, 367 Panama Street, Green Earth Sciences Building, Stanford, CA 94305, United States

Received 14 November 2006; received in revised form 8 August 2007; accepted 10 August 2007
Available online 30 August 2007

Abstract

We present a phase-based potential ordering that is an extension of the Cascade ordering introduced by Appleyard and Cheshire [John R. Appleyard, Ian M. Cheshire, The cascade method for accelerated convergence in implicit simulators, in: European Petroleum Conference, 1982, pp. 113–122]. The proposed ordering is valid for both two-phase and three-phase flow, and it can handle countercurrent flow due to gravity and/or capillarity. We show how this ordering can be used to reduce the nonlinear algebraic system that arises from the fully-implicit method (FIM) into one with only pressure dependence. The potential-based reduced Newton algorithm is then obtained by applying Newton's method to this reduced-order system. Numerical evidence shows that our potential-based reduced Newton solver is able to converge for time steps that are much larger than what the standard Newton's method can handle. In addition, whenever standard Newton converges, so does the reduced Newton algorithm, and the number of global nonlinear iterations required for convergence is significantly reduced compared with the standard Newton's method.

Keywords: Potential ordering; Nonlinear multiphase flow; Reduced Newton; Porous media; Black oil model; Nonlinear solvers

1. Introduction

The simulation of immiscible fluid displacements in underground porous media remains an important and challenging problem in reservoir engineering. First, the governing PDEs exhibit a mixed hyperbolic–parabolic character due to the coupling between the global flow and the local transport of the different phases. In addition, rock properties such as porosity and fluid permeability are highly heterogeneous, leading to poor numerical conditioning of the resulting linear systems. Finally, fluid velocities vary greatly across the domain, with near-well regions experiencing fast flows and some far away regions experiencing almost no flow at all. These characteristics impose severe constraints on the numerical methods used in practical reservoir simulation.

* Corresponding author. Tel.: +1 650 723 9476; fax: +1 650 725 2099.
E-mail address: tchelepi@stanford.edu (H. Tchelepi).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.08.012

The simplest and most widely used model in reservoir simulation is the standard black oil model [2], which incorporates various simplifying assumptions on fluid properties. (A more detailed description will be presented in Section 2.) Despite the increasing use of compositional models, black oil simulation still accounts for the vast majority of simulations in industry. Thus, this paper will concentrate on improving the efficiency and robustness of black oil simulation.

Several time discretizations are commonly used for solving the black oil equations. The IMPES method [6] makes the saturation variables explicit in time, whereas the pressure variables are implicit in time. IMPES has a stability limit that is inversely proportional to the largest fluid velocity in the reservoir, and this limit is often too restrictive in practice. The Sequential Implicit method (cf. [2]) attempts to remedy the problem by first solving for pressure using coefficients at the old time step, then computing a new total velocity field based on the new pressure, and finally solving the transport problem implicitly based on the updated total velocity field. This amounts to decoupling the flow and transport problems and solving them separately. In the absence of capillarity, the sequential implicit method does not suffer from stability problems like IMPES. However, since the flow and transport problems are actually coupled, decoupling them introduces splitting errors (and hence mass-balance errors) that are proportional to the length of the time step. Finally, the fully-implicit method (FIM) makes both the saturation and pressure variables implicit in time. This discretization can be shown to be unconditionally stable [2], and is the preferred discretization in most practical applications. Unfortunately, FIM requires the solution of a large, coupled nonlinear system of algebraic equations, so an efficient simulator must be able to solve such systems quickly and reliably. The goal of this paper is to use reordering techniques and a reduced-order Newton's method to speed up the solution of such nonlinear systems of equations.

The rest of the paper is organized as follows. In Section 2, we describe the black oil model and the spatial discretization used. Section 3 describes work related to our approach. In Section 4 we present in detail an ordering scheme that forms the basis of our solution algorithm. Section 5 shows how to use this ordering to obtain a reduced-order Newton method, which can be used to solve the global problem. Finally, in Section 6 we present some numerical examples that illustrate the effectiveness of this technique.

2. Governing equations and numerical method

2.1. Two-phase model description

Given two immiscible fluid phases w and o (water and oil), the flow of each phase is described by conservation of mass ($p = \mathrm{w}, \mathrm{o}$)

$$\frac{\partial(\phi\rho_p S_p)}{\partial t} + \nabla\cdot(\rho_p \mathbf{u}_p) = \rho_p q_p \tag{2.1}$$

and generalized Darcy's law

$$u_p = -\frac{k_{rp}}{\mu_p}\mathbf{K}\nabla\Phi_p, \tag{2.2}$$

where ϕ is the porosity of the medium, $\mathbf{K}$ is the absolute permeability tensor, z is the depth variable; and for each phase p ($p = \mathrm{o}, \mathrm{w}$), ρ_p is the density, S_p is the saturation, $\mathbf{u}_p$ is the volumetric flux vector, q_p is the source or sink term, $k_{rp} = k_{rp}(S_p)$ is the relative permeability, μ_p is the phase viscosity, $\Phi_p = p_p - \gamma_p z$ is the phase potential, p_p is the pressure, and γ_p is the gravitational force.

In addition, we have the algebraic relations

$$S_\mathrm{o} + S_\mathrm{w} = 1 \quad \text{(saturation constraint)} \tag{2.3}$$

$$p_\mathrm{o} - p_\mathrm{w} = P_\mathrm{cow}(S_\mathrm{w}), \quad \text{(capillary pressure constraint)} \tag{2.4}$$

which are used to eliminate S_o and p_o in (2.1) to obtain a coupled system of two nonlinear PDEs with S_w and p_w as the primary variables. This system of PDEs is supplemented with the boundary conditions

$$p_{\mathrm{w}} = p_{\mathrm{wd}} \quad \text{on } \Gamma_d \tag{2.5}$$

$$\rho_{\mathrm{w}}\mathbf{u}_{\mathrm{w}} \cdot \boldsymbol{\nu} = g_{\mathrm{wn}} \quad \text{on } \Gamma_n \tag{2.6}$$

$$\rho_{\mathrm{o}}\mathbf{u}_{\mathrm{o}} \cdot \boldsymbol{\nu} = g_{\mathrm{on}} \quad \text{on } \Gamma_n \tag{2.7}$$

and initial conditions

$$p_{\mathrm{w}}(x,0) = p_{\mathrm{w0}}(x), \quad S_{\mathrm{w}}(x,0) = S_{\mathrm{w0}}(x), \tag{2.8}$$

where the Dirichlet boundary has positive measure and $\boldsymbol{\nu}$ denotes the outward normal to the boundary.

2.2. Three-phase model description

For three-phase flow, we have an extra gas phase g. The conservation law for w and o remain the same as (2.1). However, since the gas component can dissolve into the oil phase, the conservation law for the gas component takes the form:

$$\left(\frac{\partial(\rho_{\mathrm{g}}\phi S_{\mathrm{g}})}{\partial t} + \nabla\cdot(\rho_{\mathrm{g}}\mathbf{u}_{\mathrm{g}})\right) + \left(\frac{\partial(\rho_{\mathrm{o}}\phi S_{\mathrm{o}}R_s)}{\partial t} + \nabla\cdot(\rho_{\mathrm{o}}\mathbf{u}_{\mathrm{o}}R_s)\right) = \rho_{\mathrm{g}}q_{\mathrm{g}}, \tag{2.9}$$

where $R_s = R_s(p_{\mathrm{g}})$ is the solubility ratio. The generalized Darcy's law (2.2) is valid for $p = \mathrm{w}, \mathrm{o}, \mathrm{g}$. In practical simulations, it is commonly assumed that the relative permeabilities k_{rp} have the following dependencies on saturation:

$$k_{rw} = k_{rw}(S_{\mathrm{w}}), \qquad k_{ro} = k_{ro}(S_{\mathrm{w}}, S_{\mathrm{g}}), \quad k_{\mathrm{rg}} = k_{\mathrm{rg}}(S_{\mathrm{g}}). \tag{2.10}$$

This parameterization is based on the assumption that water is the most wetting phase and gas the least wetting phase, which is valid for most reservoirs of interest (cf. [2] for more detailed explanations).

The algebraic relations become

$$S_{\mathrm{o}} + S_{\mathrm{w}} + S_{\mathrm{g}} = 1 \quad \text{(Saturation constraint)} \tag{2.11}$$

$$p_{\mathrm{o}} - p_{\mathrm{w}} = P_{\mathrm{cow}}(S_{\mathrm{w}}) \quad \text{(Capillary pressure constraints)} \tag{2.12}$$

$$p_{\mathrm{g}} - p_{\mathrm{o}} = P_{\mathrm{cog}}(S_{\mathrm{g}}) \tag{2.13}$$

Finally, the initial and boundary conditions (2.5)–(2.8) need to be augmented by

$$\rho_{\mathrm{g}}\mathbf{u}_{\mathrm{g}} \cdot \boldsymbol{\nu} = g_{\mathrm{g}n} \quad \text{on } \Gamma_n, \tag{2.14}$$

$$S_{\mathrm{g}}(x,0) = S_{\mathrm{g0}}(x). \tag{2.15}$$

2.3. Discretization and solution of linear systems

The two-phase or three-phase PDEs are discretized using a finite volume method with second-order central differencing for pressure, harmonic averaging for the absolute permeability tensor $\mathbf{K}$, and first-order upwinding for saturation-dependent coefficients. (For simplicity, we assume that $\mathbf{K}$ is a diagonal tensor with positive entries.) This yields a system of nonlinear algebraic equations, which we solve using Newton's method. Each Newton iteration requires the solution of the linear Jacobian system:

$$J(x^{(\nu)})\delta x^{(\nu)} = -R^{(\nu)}(x^{(\nu)}), \tag{2.16}$$

where $R^{(\nu)}$ is the residual at the iterate $x^{(\nu)}$. (Due to the large number of vectors and matrices appearing below, such quantities will no longer be shown in bold in the remainder of this paper.) Any direct or iterative linear solver can be used to solve (2.16), but a particularly effective approach is to use an iterative method such as GMRES [12] with two-stage preconditioning of the form:

$$M_{1,2}^{-1} = M_1^{-1} + M_2^{-1}(I - AM_1^{-1}). \tag{2.17}$$

The first stage preconditioner M_1 is a CPR reduction [13], which involves forming the IMPES pressure equation and solving it with an elliptic solver such as algebraic multigrid. The second stage M_2^{-1} typically uses a

local preconditioner such as ILU(0). Note that the performance of ILU preconditioners is sensitive to the ordering of variables and unknowns; such ordering effects have been considered in [7].

3. Related work

The main theme of this paper is to reorder the equations and variables in such a way that allows the saturation variables to be solved one at a time. A related approach, called the Cascade method, was proposed by Appleyard and Cheshire [1] as an acceleration scheme for the basic Newton's method. A brief description of the method follows.

Suppose we have an n_p-phase model ($n_p = 2$ or 3), in which we discretize the domain into N gridblocks. The first step in the Cascade method is the same as the ordinary Newton's method: namely, we linearize the n_pN conservation equations and solve the n_pN-by-n_pN linear system $J(x^{(\nu)})\delta x^{(\nu)} = -R^{(\nu)}(x^{(\nu)})$ for $\delta x^{(\nu)}$. Next, we apply a linear update to pressure variables p_o only, leaving the saturations intact for the time being. Using this new pressure field, we update the potential for each phase, and then we order the cells from the highest potential to the lowest. This is the order in which the Cascade sweep should be performed. Note that there is a choice in the ordering, since the potential sequence can be different for each phase. Appleyard and Cheshire suggest that one Cascade sweep be done for the potential sequence of each phase, although the method was only demonstrated for a two-phase flow problem.

Each Cascade sweep requires the solution of N single-cell problems, where N is the number of cells in the grid. For a two-phase problem, a single-cell problem has the form

$$\begin{aligned} f_o(S_w, p_o) &= \frac{1}{\Delta t}\Delta M_o(S_w, p_o) + \mathrm{FO}_o(S_w, p_o) - \mathrm{FI}_o - q_o = 0 \\ f_w(S_w, p_o) &= \frac{1}{\Delta t}\Delta M_w(S_w, p_o) + \mathrm{FO}_w(S_w, p_o) - \mathrm{FI}_w - q_w = 0 \end{aligned} \tag{3.1}$$

where ΔM_p is the accumulation of phase p, FO_p and FI_p are the outward and inward fluxes of phase p respectively, and q_p are the well terms. For a three-phase problem, we would have three such equations. We assume that the inward fluxes are known and independent of the values of S_w and p_o at the cell, which is valid provided that all neighboring cells at a higher potential have been processed, and there is no countercurrent flow. We now have a system of two nonlinear equations in two unknowns, which can be efficiently solved for S_w and p_o. The computed S_w and FO_p are retained and will be used in subsequent single-cell problems, whereas the p_o are discarded. Fig. 1 outlines one step of the cascade method.

Consider a one-dimensional model problem with

- incompressible flow,
- an injection boundary condition on the left,
- a pressure boundary condition on the right, and
- no countercurrent flow (e.g. horizontal reservoir with no capillarity).

1 Form the full Jacobian J, evaluated at (S^k, p_o^k);
2 Solve $J\begin{bmatrix}\delta S^k \\ \delta p_o^k\end{bmatrix} = r^k$;
3 Compute $p_o^{k+1} = p_o^k + \delta p_o^k$;
4 Reorder the cells so that $p_{o,i} \geq p_{o,j}$ whenever $i > j$;
5 **For** $i = 1, \ldots, N$:
6 Solve (3.1) at cell i for $S_{w,i}$ and $p_{o,i}$;
7 Update $S_{w,i}^{k+1}$ using the value from line 6;
8 Compute outward fluxes $FO_p(S_w, p_o)$ for subsequent i;
9 **end for**

Fig. 1. One iteration of the Cascade method [1].

It can be shown that the Cascade method converges to the solution in two iterations for this problem (see [9] for a proof). However, this ceases to be true in the presence of countercurrent flow or in multiple dimensions. Also, the formulation may break down if the phase potential chosen to order the cells contains local minima; in this case, the cell whose potential is at a local minimum will lack an outward flux term FO_p, so it would be impossible to satisfy mass balance for both phases no matter what S_w and p_o is. This is an important drawback because in practical applications it is usually impossible to guarantee the lack of local minima in the pressure field when the solution has not converged, especially when the initial guess is poor. In the next two sections we will show how a potential-based formulation can be used in the multidimensional case to handle gravity and capillarity systematically.

4. Potential-based ordering

In this section, we present an ordering of equations and unknowns that allows us to solve for saturation one unknown at a time, even in multiple dimensions and/or in the presence of gravity and countercurrent flow. First, we will explain how to construct this ordering in the absence of countercurrent flow; in this case the ordering will coincide with the Appleyard and Cheshire Cascade ordering. We will then extend the ordering to treat countercurrent flow due to gravity, and finally we will show how to deal with capillarity.

4.1. Cocurrent flow

Consider the two-phase model outlined in Section 2. In the absence of gravity and capillary forces, all phases will be flowing in the same direction, which is given by the negative pressure gradient $-\nabla p$ (i.e. from high to low pressure). Thus, in the finite volume discretization, the flux term between cells i and l,

$$F_{il} = \begin{cases} K \cdot \frac{k_{\mathrm{rp}}(S_l)}{\mu_p} \frac{p_l - p_i}{\Delta x}, & p_l \geqslant p_i \\ K \cdot \frac{k_{\mathrm{rp}}(S_i)}{\mu_p} \frac{p_l - p_i}{\Delta x}, & p_l \leqslant p_i \end{cases} \tag{4.1}$$

depends only on the saturation of the upstream cell. Suppose we reorder the cells such that they appear in decreasing order of pressure, i.e. $p_i \geqslant p_j$ whenever $i < j$. Then for all j, the component conservation equations for cell j depend only on saturations S_i with $i \leqslant j$. Thus, we can rearrange the system of nonlinear equations to the form:

$$\begin{aligned} &f_{c1}(S_1, p_1, \ldots, p_N) = 0 \\ &f_{c2}(S_1, S_2, p_1, \ldots, p_N) = 0 \\ &\qquad\vdots \\ &f_{cN}(S_1, S_2, \ldots, S_N, p_1, \ldots, p_N) = 0 \end{aligned} \tag{4.2}$$

where $c = \mathrm{o}, \mathrm{w}$ are the oil and water components respectively. Notice how the saturation part of the equations becomes "triangular". Thus, if we have the exact pressure solution $p_1, \ldots, p_N$, we can perform a "forward substitution" and solve a series of single-variable nonlinear equations to obtain the saturations $S_1, \ldots, S_N$. We remark that the triangularity carries over to the Jacobian matrix, which now has the form

$$J = \begin{array}{c} \begin{matrix} S_\mathrm{w} & p \end{matrix} \\ \begin{bmatrix} J_{\mathrm{ww}} & J_{\mathrm{wp}} \\ J_{\mathrm{ow}} & J_{\mathrm{op}} \end{bmatrix} \end{array} \begin{array}{l} \text{water equation} \\ \text{oil equation} \end{array} \tag{4.3}$$

where J_{ww} is lower triangular.

In the three-phase case, we have two saturation variables per cell, which we can choose as S_w and S_o without loss of generality. Since the black oil model assumes that k_{rw} depends solely on S_w, the above construction can be used to order the water equations. Now k_{ro} depends on both S_w and S_o, but we can maintain

triangularity by writing all the water equations first before writing the oil and gas equations. The nonlinear system then looks like

$$\begin{aligned}
&f_{w1}(S_{w1}, p_1, \ldots, p_N) = 0\\
&f_{w2}(S_{w1}, S_{w2}, p_1, \ldots, p_N) = 0\\
&\qquad\vdots\\
&f_{wN}(S_{w1}, \ldots, S_{wN}, p_1, \ldots, p_N) = 0\\
&f_{o1}(S_{w1}, \ldots, S_{wN}, S_{o1}, p_1, \ldots, p_N) = 0\\
&\qquad\vdots\\
&f_{oN}(S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}, p_1, \ldots, p_N) = 0\\
&\text{and}\quad f_{gi}(S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}, p_1, \ldots, p_N) = 0, \quad i = 1, \ldots, N.
\end{aligned} \tag{4.4}$$

In this case the corresponding Jacobian would have the form

$$J = \begin{array}{c} \begin{array}{ccc} S_w & S_o & p \end{array} \\ \left[\begin{array}{cc|c} J_{ww} & & J_{wp} \\ J_{ow} & J_{oo} & J_{op} \\ \hline J_{gw} & J_{go} & J_{gp} \end{array}\right] \end{array} \begin{array}{l} \text{water equation} \\ \text{oil equation} \\ \text{gas equation} \end{array} \tag{4.5}$$

with J_{ww} and J_{oo} lower triangular, which implies the entire upper-left block is lower triangular. Note that J_{ow} will also be lower triangular, since all phases have the same upstream direction. However, this fact is not needed to justify solving for S_w and S_o using forward substitution.

4.2. Countercurrent flow due to gravity

In the presence of gravity, buoyancy forces can cause different phases to flow in opposite directions. The upstream direction for each phase p is determined by the sign of $(\Phi_{p,i} - \Phi_{p,l})$, where

$$\Phi_{p,i} = p_i - \gamma_p z_i \tag{4.6}$$

is the phase potential at cell i, z_i is the depth of the cell, and γ_p is the specific gravity of phase p. Despite possible differences in upstream directions, we are interested in maintaining the triangular forms shown in (4.2) and (4.4) (and equivalently (4.3) and (4.5)). For two-phase flow, one can simply use Φ_w for ordering, since one only needs J_{ww} (and not J_{ow}) to be triangular. For three-phase flow, we need both J_{ww} and J_{oo} to be lower triangular. Clearly, no single cell ordering can accomplish this; we will need to order the water and oil phases separately. The trick is to exploit the relative permeability dependencies (2.10) in such a way that triangularity is preserved.

Unlike the cocurrent flow case, we can no longer align variable/equation ordering with cell ordering. Thus, in the sequel, subscripts (such as k in $\Phi_{p,k}$) always denote the value of the scalar field (in this case, the potential of phase p) at cell k *in the natural ordering*, since we will concentrate on ordering the equations and unknowns, rather than the cells themselves. Let $\sigma_1, \ldots, \sigma_N$ and $\tau_1, \ldots, \tau_N$ be permutations such that

$$\Phi_{w,\sigma_i} \geqslant \Phi_{w,\sigma_j} \quad \text{whenever } i < j, \tag{4.7}$$

$$\Phi_{o,\tau_i} \geqslant \Phi_{o,\tau_j} \quad \text{whenever } i < j. \tag{4.8}$$

In other words, if cell k is such that $\Phi_{w,k} > \Phi_{w,l}$ for any other l, then $\sigma_1 := k$. Suppose we order all the water equations and variables first using the σ ordering, and then order the oil equations and variables using the τ ordering. The nonlinear system then looks like

$$
\begin{aligned}
&f_{\mathrm{w},\sigma_1}(S_{\mathrm{w},\sigma_1}, p_1, \ldots, p_N) = 0\\
&f_{\mathrm{w},\sigma_2}(S_{\mathrm{w},\sigma_1}, S_{\mathrm{w},\sigma_2}, p_1, \ldots, p_N) = 0\\
&\qquad\qquad \vdots\\
&f_{\mathrm{w},\sigma_N}(S_{\mathrm{w},\sigma_1}, \ldots, S_{\mathrm{w},\sigma_N}, p_1, \ldots, p_N) = 0\\
&f_{\mathrm{o},\tau_1}(S_{\mathrm{w},\sigma_1}, \ldots, S_{\mathrm{w},\sigma_N}, S_{\mathrm{o},\tau_1}, p_1, \ldots, p_N) = 0\\
&\qquad\qquad \vdots\\
&f_{\mathrm{o},\tau_N}(S_{\mathrm{w},\sigma_1}, \ldots, S_{\mathrm{w},\sigma_N}, S_{\mathrm{o},\tau_1}, \ldots, S_{\mathrm{o},\tau_N}, p_1, \ldots, p_N) = 0\\
&\text{and} \quad f_{gi}(S_{\mathrm{w},\sigma_1}, \ldots, S_{\mathrm{w},\sigma_N}, S_{\mathrm{o},\tau_1}, \ldots, S_{\mathrm{o},\tau_N}, p_1, \ldots, p_N) = 0, \quad i = 1, \ldots, N.
\end{aligned}
\tag{4.9}
$$

Now consider the pattern of the corresponding Jacobian matrix. Clearly, J_{ww} is still lower triangular because of (4.7), and J_{oo} is lower triangular because of (4.8). The only effect of countercurrent flow is that J_{ow} will no longer be lower triangular, because the S_{w} are not arranged in decreasing Φ_{o} order. However, as long as the upper-left block in (4.5) is lower triangular, we can use forward substitution to solve for S_{w} and S_{o} once the pressures are known.

4.3. Capillarity

So far, in the absence of capillary effects, the saturation dependence in each equation is purely upstream; thus, for a given phase, saturations downstream from cell i do not appear in equation i. In contrast, equation i involves phase pressures from all neighboring cells, be they upstream or downstream from cell i. Since we can only choose one phase pressure as a primary variable, the other phase pressures must be expressed as

$$p_q = p_p + P_{\mathrm{cpq}}(S), \tag{4.10}$$

where p_p is the primary pressure and p_q is the pressure of another phase. Thus, when capillarity is present, we must choose our primary variables carefully to avoid introducing downstream dependence on saturation that cannot be removed by simply the reordering equations and unknowns. Choosing p_{w} as the primary pressure variable will allow us to maintain the triangularity in the upper-left block of (4.5). Observe that choosing p_{g} will cause the water equations to depend on S_{o}, since $p_{\mathrm{w}} = p_{\mathrm{g}} - P_{\mathrm{cog}}(S_{\mathrm{g}}) - P_{\mathrm{cow}}(S_{\mathrm{w}})$ and $S_{\mathrm{g}} = 1 - S_{\mathrm{w}} - S_{\mathrm{o}}$. This would completely destroy the triangularity of the block. If we instead choose p_{o}, then there will be no S_{o} dependence, but there will be both upstream and downstream dependence on S_{w} due to $p_{\mathrm{w}} = p_{\mathrm{o}} - P_{\mathrm{cow}}(S_{\mathrm{w}})$, which is undesirable. Thus, the only choice that leaves the water equation intact is p_{w}.

It remains to check that J_{oo} is still lower triangular when p_{w} is used. We have

$$p_{\mathrm{o}} = p_{\mathrm{w}} + P_{\mathrm{cow}}(S_{\mathrm{w}}), \tag{4.11}$$

which means we will introduce downstream dependence on S_{w}, but not on S_{o}. Hence, the J_{ow} block will now contain downstream terms, but the J_{oo} block remains unchanged. Thus, the upper-left block remains triangular, as before. The same analysis carries over in the nonlinear equation (4.9). Table 1 summarizes the ordering strategies for black oil models with different numbers of phases. Note that the gas equations, whenever they are present, are always the last to appear in the system. This is because the gas component exists in both the oil and gas phases, so no ordering can produce the required triangular forms when countercurrent flow is present.

Table 1
Ordering strategies for different black oil models

Model	Component ordering	Cell ordering		Primary pressure
		Water	Oil	
Two-phase, oil–water	Water/oil	Φ_{w}	*	p_{w}
Two-phase, gas–water	Water/gas	Φ_{w}	*	p_{w}
Two-phase, oil–gas	Oil/gas	*	Φ_{o}	p_{o}
Three-phase	Water/oil/gas	Φ_{w}	Φ_{o}	p_{w}

4.4. Remarks on implementation

In order to produce cell orderings that satisfy (4.7) and (4.8), it is not necessary to sort the potentials in decreasing order. Instead, consider the directed graph $G = (V, E)$ where the nodes V are the cells and the edges E are such that (i, j) is an edge whenever i and j are neighbors and $\Phi_i > \Phi_j$ or $\Phi_i = \Phi_j$ and $i > j$. Then a topological ordering of this graph will yield an ordering consistent with (4.7) or (4.8) depending on the potential Φ used. The running time of this operation is $O(N)$, which is asymptotically faster than sorting ($O(N \log N)$).

We also remark that in most simulations, the flow directions do not change very often, so it may not be necessary to compute this ordering at every time step. For instance, we can compute the potential ordering only at the beginning of a time step. At each subsequent Newton iteration, we can simply verify the validity of the ordering, and only recompute it when the submatrix ceases to be triangular.

5. A reduced-order Newton method

In this section, we use the potential-based ordering introduced in Section 4 to reformulate the mass-balance equations into a system of smaller size that involves pressure variables only. The Implicit Function Theorem (cf. [11]) plays a central role in the formulation. We first describe the algorithm that arises when Newton's method is applied to the reduced system.

5.1. Algorithm description

For notational convenience, we rewrite (4.9) by splitting the equations into two blocks: the first block $F_s = 0$ contains all the water and oil equations, and the second block $F_g = 0$ contains all the gas equations. Similarly, we denote the vector of all saturation variables (S_{wi} and S_{gi}, $i = 1, \ldots, N$) by S, and the vector of pressure variables by p. Then (4.9) becomes

$$\begin{cases} F_s(S,p) = 0 \\ F_g(S,p) = 0, \end{cases} \tag{5.1}$$

and the corresponding Jacobian J in (4.5) becomes

$$J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix}, \tag{5.2}$$

where

$$\begin{aligned} F_s &= [f_{w1}, \ldots, f_{wN}, f_{o1}, \ldots, f_{oN}]^T, \\ F_g &= [f_{g1}, \ldots, f_{gN}]^T, \\ S &= [S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}]^T, \\ p &= [p_{w1}, \ldots, p_{wN}]^T, \end{aligned}$$

and

$$J_{ss} = \partial F_s/\partial S, \quad J_{sp} = \partial F_s/\partial p, \quad J_{gs} = \partial F_g/\partial S, \quad J_{gp} = \partial F_g/\partial p.$$

It can be shown that J_{ss} is non-singular as long as the monotonicity condition $dk_{rp}/dS_p \geqslant 0$ is valid for $p = o,w$ (see Appendix A for the proof). For $k_{rw} = k_{rw}(S_w)$ (which is usually obtained from experimental data), monotonicity is almost always satisfied, but the situation is less clear for $k_{ro} = k_{ro}(S_w, S_g)$, since the latter is usually obtained by interpolating data from oil–water and oil–gas experiments. Certain methods of interpolation, such as Stone I and Stone II, yield monotonic k_{ro} under mild conditions (see Appendix B), but this is not always the case for other methods (e.g. the segregation model). In this work it is assumed that k_{ro} is a monotonically increasing function of S_o when S_w is fixed, which would ensure the invertibility of J_{ss}..

Consequently, since $F_s(S, p)$ has a triangular structure with respect to saturation, one can solve for S one unknown at a time if p is given. In addition, the implicit function theorem guarantees that if $F_s(S_0, p_0) = 0$ and

$\partial F_s/\partial S$ is non-singular at (S_0, p_0), then there exists a neighborhood U of p_0 and a unique differentiable function $S = S(p)$ such that $S(p_0) = S_0$ and $F_s(S(p), p) = 0$ for all $p \in U$. In other words, we can use F_s as a constraint to define saturation as a function of pressure, and substitute it into the remaining equations F_g. Thus, we obtain

$$F_g(S(p), p) = 0, \tag{5.3}$$

which we need to solve for the pressure p. If we use Newton's method to solve (5.3), the Jacobian matrix becomes

$$J_{\text{reduced}} = \frac{\partial F_g}{\partial S}\frac{\partial S}{\partial p} + \frac{\partial F_g}{\partial p} \tag{5.4}$$

$$= J_{gs}\frac{\partial S}{\partial p} + J_{gp}. \tag{5.5}$$

Now $\partial S/\partial p$ is given by the implicit function theorem: $F_s(S(p),p) \equiv 0$ implies

$$\frac{\partial F_s}{\partial S}\frac{\partial S}{\partial p} + \frac{\partial F_s}{\partial p} = 0, \tag{5.6}$$

which gives

$$\frac{\partial S}{\partial p} = -\left(\frac{\partial F_s}{\partial S}\right)^{-1}\frac{\partial F_s}{\partial p}, \tag{5.7}$$

which can be written as $\partial S/\partial p = -J_{ss}^{-1}J_{sp}$. Thus, the reduced Jacobian matrix is

$$J_{\text{reduced}} = J_{gp} - J_{gs}J_{ss}^{-1}J_{sp}, \tag{5.8}$$

which is precisely the Schur complement of (5.2) with respect to pressure. Fig. 2 summarizes the algorithm used to solve the reduced system. Notice that the only difference between the algorithm in Fig. 2 and Newton's method applied to the full problem is the way we compute S^{k+1}. In the full method, we set $S^{k+1} = S^k + \delta S^k$; in the reduced method, S^{k+1} is updated nonlinearly by solving the constraint equations $F(S^{k+1},p^{k+1}) = 0$, exploiting the special triangular structure of J_{ss}. Also note that since this is just the usual Newton's method applied to a reduced problem, *convergence is locally quadratic*.

5.2. Sequential updating of the saturations

The algorithm in Fig. 2 requires the solution of $F_s(S^{k+1}, p_w^{k+1}) = 0$ for S^{k+1} at every step. Using the potential ordering in Section 4, we can triangularize the constraint equations to obtain the system (4.9). Thus, given the pressure values $p_1, \ldots, p_N$, we first solve $f_{w1} = 0$ for S_{w1}. Then, using this S_{w1} we can now solve $f_{w2} = 0$ for S_{w2}, and so on until we obtain all saturation values. Thus, solving $F_s(S^{k+1}, p_w^{k+1}) = 0$ reduces to solving $(n_p - 1)N$ nonlinear scalar equations one at a time (where n_p is the number of fluid phases). A wide variety of reliable univariate solvers are available to deal with the single-cell problems. One such choice is the van Wijngaarden–Dekker–Brent Method [3], which combines bisection with inverse quadratic interpolation to obtain superlinear convergence. This is a derivative-free algorithm, which means only function values are required,

```
1  while |F_g(S(p_w^k), p_w^k)| > tol , do
2      Form the full Jacobian J = [J_ss  J_sp; J_gs  J_gp] , evaluated at (S(p_w^k), p_w^k) ;
3      Solve (J_gp − J_gs J_ss^{-1} J_sp) δp^k = −r^k ;
4      Compute p_w^{k+1} = p_w^k + δp^k ;
5      Update S^{k+1} = S(p_w^{k+1}) nonlinearly by solving F_s(S^{k+1}, p_w^{k+1}) = 0,
6          one variable at a time in potential ordering ;
7      k := k + 1
8  end
```

Fig. 2. Algorithm for solving the reduced system (5.3).

although an initial guess based on the solution of the ordinary Newton step can be used to accelerate convergence. In a reasonably efficient implementation, each function evaluation should only require a few floating-point operations. As shown in Section 6, the extra cost of the single-cell nonlinear solves is usually offset by a reduction of the number of global Newton steps. The nonlinear updates can be performed more efficiently if more sophisticated zero finders are used.

5.3. Solving the Schur complement system

There are two ways to solve the Schur complement system

$$J_{\text{reduced}}\delta p = -r. \tag{5.9}$$

The first way is to notice that one can solve the equivalent system

$$\begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix} \begin{bmatrix} \delta S \\ \delta p \end{bmatrix} = \begin{bmatrix} 0 \\ -r \end{bmatrix}. \tag{5.10}$$

Krylov subspace methods (such as GMRES) can be used, and effective preconditioners (such as CPR [13]) are available. A second way is to apply the Krylov method directly to the Schur complement system. In this approach, matrix–vector multiplication by J_{reduced} would have the same cost as multiplication by the full matrix, because J_{ss} is lower triangular, so that multiplication by J_{ss}^{-1} is simply a forward substitution. In terms of preconditioning, one can use an *induced* preconditioner based on the full system by letting

$$M_{\text{reduced}}^{-1} = RM_{\text{full}}^{-1}R^{\text{T}}, \tag{5.11}$$

where M_{full}^{-1} is the preconditioner for the full system, and $R = [0\ I]$ is the restriction operator to the pressure variables. In other words, a preconditioning step for the reduced system $y = M_{\text{reduced}}^{-1}x$ consists of the following steps:

(1) Pad the vector x with zeros to form $\hat{x} = (\,0_{(n_p-1)N} \quad x\,)$.
(2) Compute $\hat{y} = M_{\text{full}}^{-1}\hat{x}$.
(3) Let y be the portion of $\hat{y}$ corresponding to pressure variables, i.e. retain only the last N elements of $\hat{y}$.

One advantage of applying the Krylov method to the Schur complement system rather than the full system is that the resulting Krylov vectors are only of length N rather than length $n_{\text{p}}N$, where n_{p} is the number of fluid phases. This greatly reduces storage and orthogonalization costs in methods such as GMRES, so that more Krylov steps can be taken before restarting.

In fact, the Schur complement reduction can be used even if the nonlinear constraint equations are not exactly satisfied. This could happen if the initial pressure guess is so poor that some of the residual constraints in the reduced Newton cannot be satisfied. In that case we would have

$$\begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix} \begin{bmatrix} \delta S \\ \delta p \end{bmatrix} = -\begin{bmatrix} r_s \\ r_{\text{g}} \end{bmatrix} \tag{5.12}$$

But this is equivalent to solving

$$J_{\text{reduced}}\delta p = -(r_{\text{g}} - J_{gs}J_{ss}^{-1}r_s) \tag{5.13}$$

which has the same form as (5.9), so we get the same computational savings as before.

To test the efficiency of the potential-based reduced Newton algorithm, we implement it inside GPRS, which was developed by Cao [4] in 2002 as general-purpose research simulator. It is used by Stanford University's SUPRI-B and SUPRI-HW research groups, as well as other research groups and companies for their in-house research. By implementing our algorithm in GPRS we can guarantee that all the property calculations and convergence checks are identical for both the standard and reduced Newton methods. We can also ensure our reference point is indeed the basic Newton method, rather than a version adorned with various heuristics. Consequently, all foregoing comparisons between the standard and reduced Newton methods are generated by GPRS.

6. Numerical examples

6.1. 1D example with gravity

To demonstrate that the potential-based reduced Newton algorithm does indeed work in the presence of countercurrent flow, we first test it on a simple pseudo-1D example. The reservoir is discretized using $10 \times 1 \times 100$ cells in the x, y and z directions, respectively, with $D_x = 10$ ft, $D_y = 50$ ft and $D_z = 4$ ft. A uniform porosity ($\phi = 0.3$) and permeability ($k_x = k_y = k_z = 758$ md) are used. Water is injected across the top layer at a rate of 213.6 bbl/day (0.002 pore volumes per day) and a production well is completed across the bottom layer and operates at a BHP of 500 psi. The densities of water and oil at standard conditions are 64 lb/cu.ft. and 49 lb/cu.ft., respectively, and the viscosities are $\mu_o = 1.0$ cp, $\mu_w = 0.3$ cp. The fractional flow curve for this problem is shown in Fig. 3. We see that flow is cocurrent for $0 \leqslant S_w \leqslant 0.38$, and countercurrent for $0.38 \leqslant S_w \leqslant 1$. We test our algorithm for initial water saturations $S_{wi} = 0.0, 0.1, \ldots, 0.9$. In each case, the simulation steps through $T = 1, 3, 7, 15, 30, 45, 60$ days (1 day = 0.002 pore volumes), and afterwards the time-step size is fixed at $\Delta T = 20$ days until $T = 300$ days is reached, for a total of 21 steps. Table 2 shows the results for the standard and reduced Newton algorithms. We see that reduced Newton does not need to cut any time steps to achieve convergence, whereas Standard Newton must cut the time step multiple times in four

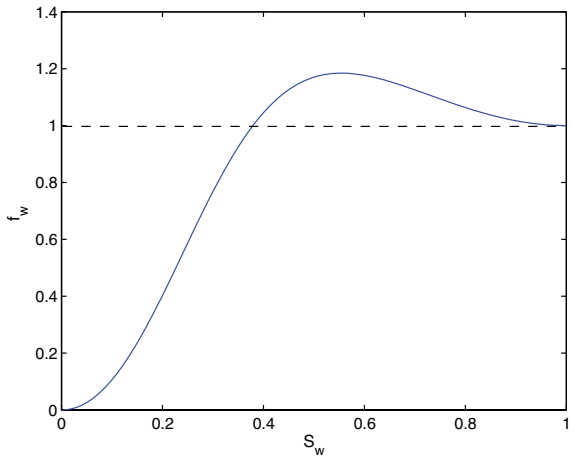

Fig. 3. Fractional flow f_w for the 1D gravity example.

Table 2
Convergence history for 1D flood with different initial water saturations

S_{wi}	Standard			Reduced		
	Timesteps	Newtons	Cuts	Timesteps	Newtons	Cuts
0.0	26	140	5	21	61	0
0.1	21	59	0	21	58	0
0.2	21	59	0	21	58	0
0.3	21	50	0	21	49	0
0.4	21	51	0	21	58	0
0.5	21	67	0	21	81	0
0.6	22	88	2	21	85	0
0.7	24	96	6	21	90	0
0.8	23	85	3	21	84	0
0.9	21	51	0	21	65	0

For both methods: Timesteps, total number of timesteps taken to simulate up to 300 days; Newtons, number of Newton iterations (excluding iterations wasted due to time-step cuts); Cuts, number of times the algorithm must cut the time-step size by half due to non-convergence.

cases ($S_w = 0.0, 0.6, 0.7, 0.8$). Time-step cuts are very expensive, since it means that we must throw away the results of all previous iterations and start over. Moreover, the size of the next step following a time-step cut is usually set to the last successfully integrated Δt, i.e. the one reduced by the time-step cut. This can lead to a significantly smaller average time-step size for a given simulation. Thus, a more stable algorithm that avoids time-step cuts can significantly outperform one that cuts time steps frequently, especially if their convergence rates are otherwise comparable. Table 2 shows that when neither algorithm requires time-step cuts, standard Newton converges more quickly some of the time ($S_w = 0.4, 0.5, 0.9$), whereas reduced Newton is quicker at other times ($S_w = 0.1, 0.2, 0.3, 0.6$). Nonetheless, the differences in the average iteration count is less than 0.67 iterations per time step in all cases. So the convergence rates for both algorithms are comparable when no time-step cuts are needed. As we will see in later examples, the enhanced stability of reduced Newton does translate into gains in actual running time for larger problems. The primary goal of this example is to demonstrate the robustness of reduced Newton, even in the presence of strong countercurrent flow. This property is essential if the algorithm is to be used in heterogeneous reservoirs with complicated permeability/porosity fields, especially since countercurrent flow due to gravity can be important in regions where the total velocity is small.

6.2. Heterogeneous example with gravity

To demonstrate the effectiveness of reduced Newton on a large, complex heterogeneous reservoir, we test it on a water flood problem using a $2 \times 2 \times 2$ upscaling of the SPE 10 model [5]. This gives rise to a model with 141,900 grid blocks ($110 \times 30 \times 43$). A visualization of the reservoir is shown in Fig. 4. The top 18 layers of the reservoir represent a Tarbert formulation with highly variable permeabilities ranging from 4.8×10^{-3} to 1.2×10^{3} md. The bottom 25 layers consists of an Upper Ness sequence, which is highly channelized. The porosity is 0.3 throughout the reservoir. Water is injected at the center of the reservoir at 5000 bbl/day (=0.0002 pore volumes per day); four production wells are located in the four corners of the reservoir, operating at a bottom hole pressure of 4000 psi. Quadratic relative permeabilities are used with a residual saturation of 0.2 for both phases, and the viscosity ratio is 10. The rest of the parameters are the same as those in the original specification. The simulation is carried out up to $T = 500$ days, which corresponds to 0.1 pore volumes injected (PVI). For any time step, if the global nonlinear solver does not converge within 20 iterations, the iterations are stopped and the current time step is cut in half before restarting. Table 3 shows the convergence history of the standard and reduced Newton methods for an initial time step of 0.1 days. Here the time

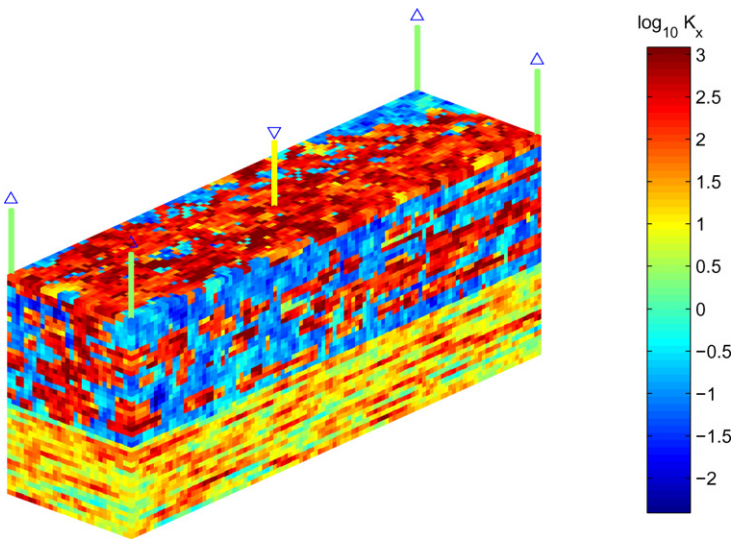

Fig. 4. Permeability field and well configuration for the upscaled SPE10 problem[5]. The reservoir is displayed upside down so that the channels in the bottom layers are clearly visible.

Table 3
Convergence history for the upscaled SPE 10 model with an initial time step of 0.1 days

Days	Standard		Reduced		CFL	%CC
	N	L	N	L		
0.1	4	18	4	17	1.8	6.2
0.3	3	17	3	17	1.9	2.4
0.7	3	18	2	12	2.1	1.1
1.5	3	19	2	14	2.5	0.7
3.1	4	26	2	15	4.0	0.5
6.3	5	32	2	16	6.7	0.5
10	4	26	2	15	11.1	0.5
20	6	45	3	27	23.9	0.5
35	4	32	3	27	35.2	0.5
50	3	27	2	19	33.2	0.5
70	4	35	3	27	35.1	0.6
90	4	33	3	28	35.6	0.6
110	4	37	3	30	52.9	0.6
140	4	41	3	34	112.1	0.6
170	4	39	2	21	102.8	0.7
200	4	35	2	21	145.3	0.7
230	3	33	2	22	129.1	0.7
260	3	33	2	22	132.0	0.8
290	3	30	2	21	132.3	0.8
320	3	31	2	21	119.6	0.8
350	3	30	2	19	109.5	0.8
380	3	30	2	20	116.7	0.9
410	3	31	2	20	112.0	0.9
440	3	30	2	19	114.9	0.9
470	3	28	2	19	108.1	1.0
500	3	29	2	19	146.3	1.0
Total	93	785	61	542		
Running time (s)	728.6		560.6			

N, number of nonlinear (Newton) iterations; L, number of Linear (CPR) solves; CFL, maximum CFL number in the reservoir; %CC, percentage of cell interfaces that experience countercurrent flow.

stepping is gentle enough that standard Newton does not need to cut time steps in order to achieve convergence. We see that reduced Newton takes fewer iterations than standard Newton to converge, and that the running time decreases from 728.6 s to 560.6 s. Thus, the savings from reducing the number of Newton iterations are more than enough to offset the cost of univariate solves.

Next we specify an initial time step of 1 day and track the number of Newton iterations required to converge. Fig. 5 shows the results. We see that reduced Newton converges for the first time step in 9 iterations, whereas Standard Newton does not converge and needs to cut the time step twice to converge with an initial time step of 0.25 days. Beyond the first time step, reduced Newton always takes fewer iterations to converge than its standard counterpart, and the iteration count does not exhibit the large variations that Standard Newton does at the beginning.

6.3. Large heterogeneous example

Since the cost of the single cell solves only scales linearly with the problem size, we expect that the savings afforded by the potential-based reduced Newton method will become even more evident in large heterogeneous examples, where the computational cost is dominated by the solution of linear systems. We demonstrate this by simulating the full SPE 10 problem ($60 \times 220 \times 85 = 1.12$ million grid blocks) and with the variable porosity field as specified in [5]. The published relative permeabilities and fluid properties are used, except that the formation volume factor B_o and the density ρ_o are taken to be the same as the published B_w and ρ_w. The injection rate is 5000 bbl/day (0.000366 pore volumes per day). The simulation runs until $T = 2000$ days (PVI = 0.732). Three time-stepping strategies are used:

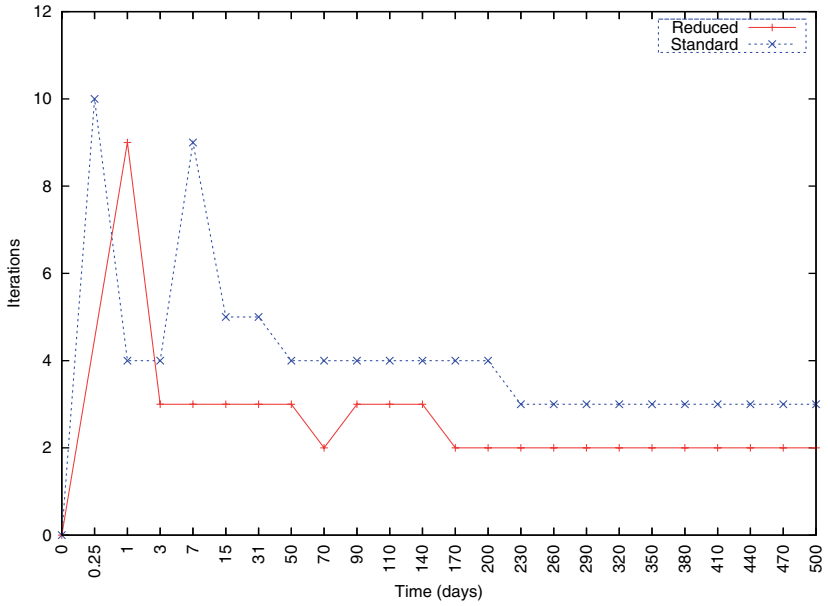

Fig. 5. Convergence history for the upscaled SPE 10 model with an initial time-step of 1 day.

- *Short time steps*: $T = 0.01, 0.03, 0.07, 0.15, 0.31, 0.63, 1, 3, 7, 15, 31, 63, 90, 120, 150, 180, 220, 260, 300$ days. After 300 days, $\Delta T = 50$ days (0.0183 pore volumes) until $T = 2000$ days is reached.
- *Long time steps*: $T = 0.01, 0.31, 1, 7, 31, 90, 150, 220, 300$ days. After 300 days, $\Delta T = 100$ days (0.0366 pore volumes) until $T = 2000$ days is reached.
- *Huge time steps*: $T = 0.01, 0.31, 1, 7, 31, 90, 200$ days. After 200 days, $\Delta T = 500$ days (0.183 pore volumes) until $T = 2000$ days is reached.

As before, the time step is cut in half if the global nonlinear solver does not converge within 20 iterations. Table 4 summarizes the runs for both the standard and reduced Newton algorithms, and Fig. 6 compares the convergence histories of standard and reduced Newton for the long time step case. We observe that reduced Newton can easily handle the "long" and "huge" time step cases, whereas standard Newton needs to cut time steps multiple times in order to achieve convergence. This results in a significant number of wasted linear solves and a serious degradation in performance. In fact, we could not run standard Newton for the huge time step case due to the large number of time step cuts; thus, taking too large a time step in standard Newton actually makes the simulation slower, whereas the opposite is true for reduced Newton. Indeed, reduced Newton with long or huge time steps runs in less than 60% of the time required by standard Newton with either time-step-

Table 4
Summary of runs for the full SPE 10 problem

	Standard		Reduced		
	Short Δt	Long Δt	Short Δt	Long Δt	Huge Δt
No. of time steps	58	38	53	26	11
No. of time step cuts	6	17	0	0	0
No. of Newton steps	353	516	128	90	55
Wasted Newton steps	120	340	0	0	0
No. of linear solves	3818	6257	2271	2399	1805
Wasted linear solves	860	3934	0	0	0
Total running time (s)	24,053	37,388	16,558	14,727	10,275
Linear solves (s)	22,570	35,457	11,697	11,301	7899
Single-cell solves (s)	0	0	4194	2996	2132

"Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted due to time step cuts.

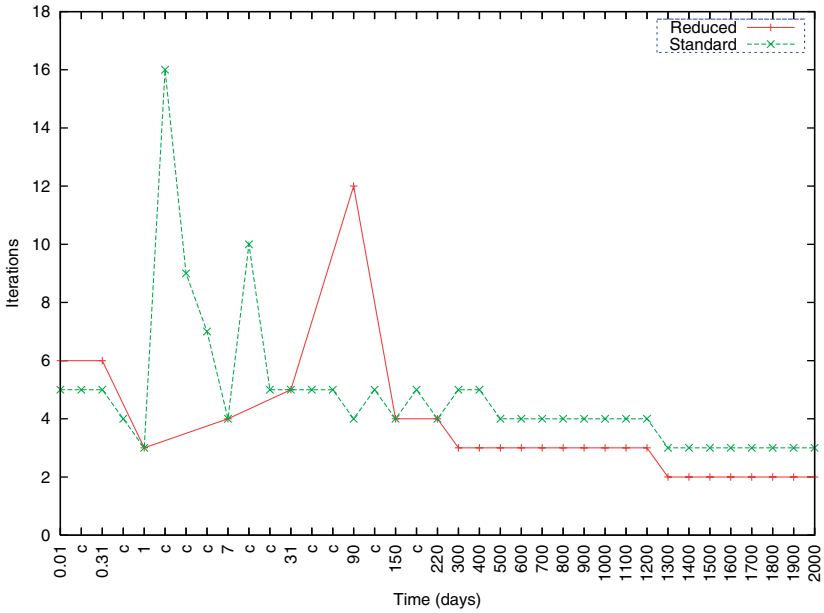

Fig. 6. Convergence history for the full SPE10 problem with long time steps. Tick marks on the x-axis labeled c correspond to intermediate time steps needed by standard Newton to achieve convergence; these steps are skipped by reduced Newton.

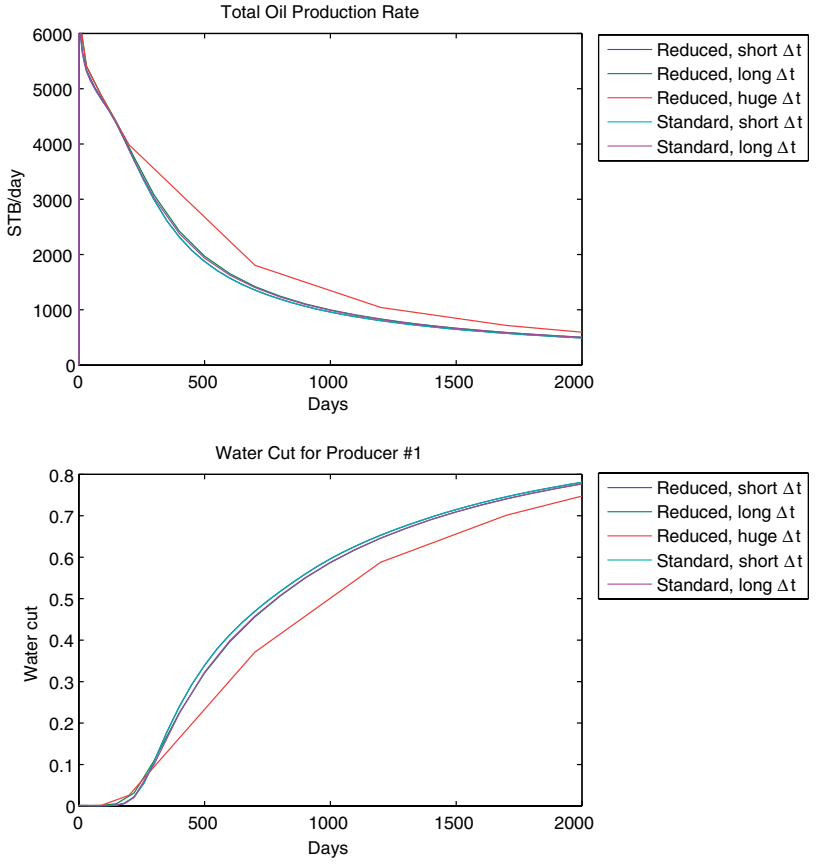

Fig. 7. Total oil production rate and water cut for the full SPE 10 problem.

ping strategy. Finally, Fig. 7 shows the oil production rate and water cut for all four simulation runs. Except for the huge time step case, the discrepancy between the solutions is insignificant. Thus, by using reduced Newton with larger time steps, we obtain substantial speedups with little or no change in solution accuracy.

6.4. 1D three-phase example with gravity

To show that the reduced formulation is applicable to three-phase flow, the algorithm is tested on a three-phase model, in which gas is injected into a reservoir initially containing a mixture of 50% oil and 50% water in every cell. This saturation is chosen to ensure that all phases are mobile, and that we have a truly three-phase

Table 5
PVT relations for all three-phase examples

P (psi)	B_o (RB/STB)	μ_o (cp)	B_w (RB/STB)	μ_w (cp)	B_g (RB/SCF)	μ_g (cp)
14.7	1.062	2.200	1.0410	0.31	0.166666	0.0080
264.7	1.061	2.850	1.0430	0.31	0.012093	0.0096
514.7	1.060	2.970	1.0395	0.31	0.006274	0.0112
1014.7	1.059	2.990	1.0380	0.31	0.003197	0.0140
2014.7	1.056	2.992	1.0350	0.31	0.001614	0.0189
2514.7	1.054	2.994	1.0335	0.31	0.001294	0.0208
3014.7	1.053	2.996	1.0320	0.31	0.001080	0.0228
4014.7	1.050	2.998	1.0290	0.31	0.000811	0.0268
5014.7	1.047	3.000	1.0258	0.31	0.000649	0.0309
9014.7	1.033	3.008	1.0130	0.31	0.000386	0.0470

Table 6
Relative permeabilities for all three-phase examples

S_w	k_{rw}	k_{row}
0.12	0	1.00
0.121	1.67E − 12	1.00
0.14	2.67E − 07	0.997
0.17	1.04E − 05	0.98
0.24	3.46E − 04	0.7
0.32	2.67E − 03	0.35
0.37	6.51E − 03	0.2
0.42	0.014	0.09
0.52	0.043	0.021
0.57	0.068	0.01
0.62	0.104	0.001
0.72	0.216	0.0001
0.82	0.400	0
1.00	1.000	0
S_g	k_{rg}	k_{rog}
0	0	1.00
0.001	0.0002	1.00
0.02	0.0033	0.997
0.05	0.0106	0.98
0.12	0.0364	0.70
0.20	0.0919	0.35
0.25	0.1459	0.20
0.30	0.2226	0.09
0.40	0.4588	0.021
0.45	0.6336	0.01
0.50	0.7449	0.001
0.60	0.8887	0.0001
0.70	0.9563	0
0.88	1.0000	0

problem. The reservoir is identical to the one used in Example 6.1. PVT data and relative permeabilities are shown in Tables 5 and 6, respectively. The gas component is assumed not to dissolve into the oil phase, (i.e. $R_{go} = 0$). The oil relative permeability is interpolated from the oil–gas and oil–water tables using the Stone I method. Gas is injected into the top layer at a rate of 100 MSCF/day (0.000768 pore volumes/day at 4000 psi), and a producer in the bottom layer is maintained at a constant pressure of 4000 psi. The production curve is shown in Fig. 8. Even though gas is highly mobile ($\mu_w/\mu_g = 11.6$, $\mu_o/\mu_g = 111.9$), breakthrough occurs relatively late (at $T = 521$ days or 0.4 pore volumes) because gas preferentially stays in the upper layers due to buoyancy. In addition, since the simulation does not start from gravity equilibrium, gravity segregation between oil and water must occur at the initial stages of the simulation. Up to 98% of cell interfaces experience countercurrent flow at some point before gas breakthrough. This accounts for the rather complicated behavior of the water and oil production curves prior to gas breakthrough. Even though this is a rather small example, we believe it captures the essence of the types of nonlinearity present in countercurrent three-phase flow, and that it provides a good test case for comparing the convergence behavior of the standard and reduced Newton algorithms. In this example, two time-stepping strategies are used:

- *Short time steps*: $T = 0.1,1,5,10$ days. After 10 days, $\Delta t = 10$ days (0.00768 pore volumes) until $T = 1000$ days.
- *Long time steps*: After an initial time step of 0.1 days, Δt is automatically chosen based on saturation and pressure changes, with a minimum of $\Delta t = 10$ days and gradually increasing until $\Delta t = 100$ days (0.0768 pore volumes).

Table 7 summarizes the runs for the standard and potential-based reduced Newton algorithms. (Running times have little meaning due to the small size of the problem, and are thus omitted.) We once again observe that reduced Newton has no problems handling both short and long time steps, whereas standard Newton

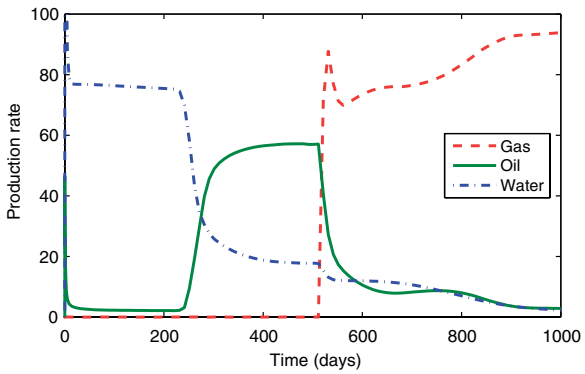

Fig. 8. Production curve for the 1D three-phase example. The units are STB/day for oil and water, and MSCF/day for gas.

Table 7
Summary of runs for 1D three-phase example with gravity

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
Number of time steps	111	74	103	26
Number of time step cuts	16	36	0	0
Number of Newton steps	888	1223	480	229
Wasted Newton steps	320	720	0	0
Number of linear solves	1763	2421	973	480
Wasted linear solves	641	1418	0	0

"Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted due to time step cuts.

needs to cut a significant amount of time steps in both cases to achieve convergence. Thus, the presence of three phases does not negatively impact the convergence behavior of reduced Newton.

6.5. 2D Heterogeneous three-phase example

We now test the reduced Newton algorithm on a three-phase example with heterogeneity. The reservoir consists of the 51st layer of the SPE 10 problem, which is a slice in the Upper Ness formation (see Example 6.3). Initially the reservoir contains a mixture of 50% oil and 50% water, and gas is injected through a well in the center at a rate of 1000 MSCF/day (0.00005 pore volumes per day). The four production wells (one in each corner) are each maintained at a bottom hole pressure of 4000 psi. The PVT and relative permeability data are the same as in Example 6.4 and are given in Tables 5 and 6. The simulation is run up to $T = 500$ days (0.025 PVI), which is much larger than the breakthrough time ($T_{BT} \approx 40$ days or 0.002 PVI). Note that the early breakthrough time is due to the extremely high mobility of the gas. Fig. 9 shows the gas saturation of the reservoir at $T = 500$ days. Two time-stepping strategies are used:

- *Short time steps*: $T = 1, 3, 7, 15, 31, 63, 100$ days. After 100 days, $\Delta t = 50$ days (0.00125 pore volumes) until $T = 500$ days is reached.
- *Long time steps*: $T = 10, 30, 60, 100$ days. After 100 days, $\Delta t = 100$ days (0.0025 pore volumes) until $T = 500$ days is reached.

Table 8 shows the performance of the standard and reduced Newton algorithms. Once again no time step cuts are required by reduced Newton, demonstrating its stability compared with the standard Newton's method. This translates to an improvement in running time for the long time step case. This example shows that the improvement obtained from reduced Newton in three-phase flow is not limited to simple 1D cases.

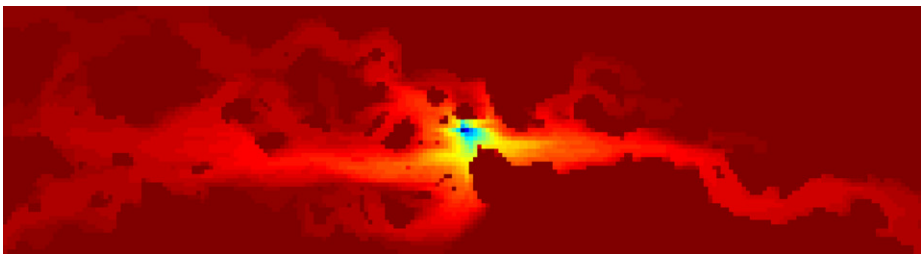

Fig. 9. Gas saturation at $T = 500$ days in the 2D heterogeneous three-phase example. Dark blue indicates 100% gas, whereas dark red indicates a cell consisting purely of liquid phases.

Table 8
Summary of runs for 2D heterogeneous three-phase example

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
Number of time steps	16	10	15	8
Number of time step cuts	1	3	0	0
Number of Newton steps	74	101	58	40
Wasted Newton steps	20	60	0	0
Number of linear solves	1264	1529	1172	881
Wasted linear solves	276	698	0	0
Total running time (s)	63.5	75.6	73.8	53.9
Linear solves (s)	53.7	66.1	50.5	37.9
Single-cell solves (s)	0	0	18.6	13.0

"Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted due to time step cuts.

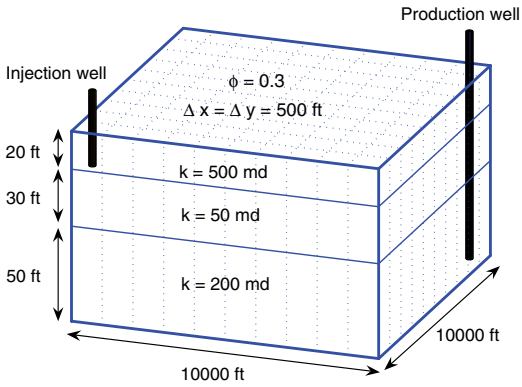

Fig. 10. Reservoir description for the 3D three-phase example.

Table 9
Summary of runs for 3D three-phase example

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
Number of time steps	18	11	17	9
Number of time step cuts	1	3	0	0
Number of Newton steps	95	117	74	57
Wasted Newton steps	20	60	0	0
Number of linear solves	1083	1624	974	838
Wasted linear solves	178	855	0	0
Total running time (s)	4.9	6.5	6.1	4.9
Linear solves (s)	3.7	5.5	3.4	2.9
Single-cell solves (s)	0	0	2.1	1.6

"Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted due to time step cuts.

6.6. 3D three-phase example

Finally, the algorithm is tested on a 3D three-phase model, in which gas is injected into a reservoir containing a mixture of 50% oil and 50% water. The reservoir ($20 \times 20 \times 3$ cells) is a 2×2 areal refinement of the one used in the SPE1 test set [10] and is shown in Fig. 10. The PVT data and relative permeabilities are the same as the two previous examples (Tables 5 and 6), and the Stone I model is used to interpolate the oil–gas and oil–water data. The gas-injection well is completed in cell (1, 1, 1) and operates at 100,000 MSCF/day (0.000073 pore volumes per day at 9000 psi); a production well, completed in cell (20, 20, 3), operates at a bottom-hole pressure of 1000 psi. The simulation is run up to $T = 5000$ days (0.365 PVI). Again because of the high gas mobility, breakthrough occurs very early ($T_{BT} \approx 100$ days or 0.0073 PVI). Since the oil and water are not in gravity equilibrium at the start of the simulation, there is significant countercurrent flow in the problem. Two time-stepping strategies are used:

- *Short time steps*: $T = 30, 100, 200, 250, 400, 600, 900$ days. After 900 days, $\Delta t = 400$ days (0.0292 pore volumes) until $T = 5000$ days.
- *Long time steps*: $T = 100, 250, 600$ days. After 600 days, $\Delta t = 800$ days (0.0584 pore volumes) until $T = 5000$ days.

Table 9 shows the performance of the standard and reduced Newton algorithms. Again we see that the reduced Newton method requires no time step cuts and fewer iterations to converge when compared to the standard Newton method.

7. Conclusions

We proposed a potential-based ordering scheme for nonlinear systems of algebraic equations that arise from a fully-implicit finite volume discretization of the standard black oil model. This ordering generalizes the Cascade ordering by Appleyard and Cheshire [1] to handle countercurrent flow due to gravity and/or capillarity, and is applicable to both two and three-phase flow. For a given pressure field, this ordering allows us to solve for the saturations of each gridblock by solving $(n_p - 1)N$ single-cell scalar equations in one variable, where n_p is the number of phases and N is the number of gridblocks.

Based on this ordering, we developed a reduced-order Newton method by implicitly defining saturation as a function of pressure. The reduced problem is then solved using Newton's method. Thus, this method can be regarded as a nonlinear analog of a Schur complement reduction. Solving the reduced Jacobian system is cheaper than solving the full problem, since the dimension of the residual space is now N rather than $n_p N$. Moreover, numerical evidence showed that the potential-based reduced Newton solver was able to converge for time steps that were much larger than what the standard Newton's method could handle. In addition, whenever standard Newton converged, the reduced Newton algorithm also converged, and usually in fewer iterations than standard Newton. This led to a significant improvement in overall solution time, especially on large heterogeneous problems such as SPE 10 [5]. We are currently investigating ways to take advantage of the potential ordering in the linear solution step. Moreover, we are working on hybridizing the potential-based Reduced Newton algorithm with standard Newton for general black oil and compositional problems.

We acknowledge the support of this research by the SUPRI-B reservoir simulation affiliates program, at Stanford University.

Appendix A. Invertibility of J_{ss}

Proposition 1. *Let the relative permeability functions k_{rw} and k_{ro} be such that $dk_{rw}/dS_w \geqslant 0$, $\partial k_{ro}/\partial S_o \geqslant 0$. Then $J_{ss} = \partial F_s/\partial S$ is non-singular.*

Proof. Since J_{ss} is a lower triangular matrix, it suffices to show that none of its diagonal entries are zero. A typical oil conservation equation for cell i is

$$F_{oi} = \frac{\phi S_{oi} b_o(p_i)}{\Delta t} + \sum_{l \text{ adjacent to } i} K_{il} H_{o,il} (\Phi_{oi} - \Phi_{ol}) + F_{cap} \tag{A.1}$$

where

$$H_{o,il} = \begin{cases} k_{ro}(S_i) b_o(p_i)/\mu_o(p_i) & \text{if } \Phi_{oi} \geqslant \Phi_{ol}, \\ k_{ro}(S_l) b_o(p_l)/\mu_o(p_l) & \text{if } \Phi_{oi} < \Phi_{ol}, \end{cases}$$

and F_{cap} denotes capillary forces that are independent of S_o. Hence

$$\frac{\partial F_{oi}}{\partial S_{oi}} = \frac{\phi b_o(p_i)}{\Delta t} + K_{il} \frac{\partial H_{o,il}}{\partial S_{oi}} (\Phi_{oi} - \Phi_{ol}). \tag{A.2}$$

The accumulation term $\phi b_w(p_i)/\Delta t$ will always be positive. The sign of the flux term depends on the upstream direction. If $\Phi_{oi} \geqslant \Phi_{ol}$, then

$$\frac{\partial H_{o,il}}{\partial S_{oi}} = \frac{b_o(p_i)}{\mu_o(p_i)} \frac{\partial k_{ro}}{\partial S_o}(S_{oi}) \geqslant 0$$

by assumption. On the other hand, if $\Phi_{oi} < \Phi_{ol}$, then $H_{o,il}$ is independent of S_{oi}, so the derivative is zero. Thus, the flux derivative will always be non-negative, which means $\partial F_{oi}/\partial S_{oi} > 0$ for all cells i. The argument for the water equations is similar. Thus, J_{ss} has positive diagonal, so it is invertible. □

Appendix B. Monotonicity of Stone's Models

Under certain mild conditions (to be specified below), the Stone I and II models (cf. [2]) can be shown to satisfy $\partial\, k_{\mathrm{ro}}/\partial S_{\mathrm{o}} \geqslant 0$, as required by Proposition 1. Note that we are only concerned with saturations inside the region

$$D = \{(S_{\mathrm{w}}, S_{\mathrm{o}}, S_{\mathrm{g}}) | S_{\mathrm{w}} \geqslant S_{\mathrm{wc}}, S_{\mathrm{o}} \geqslant S_{\mathrm{om}}, S_{\mathrm{g}} \geqslant 0, S_{\mathrm{w}} + S_{\mathrm{o}} + S_{\mathrm{g}} = 1\},$$

where S_{wc} is the connate water saturation and S_{om} is the minimum oil saturation at which oil is simultaneously displaced by water and gas. Also note that the derivative $\partial k_{\mathrm{ro}}/\partial S_{\mathrm{o}}$ is taken along the line $S_{\mathrm{w}} = \text{constant}$, so by the relation $S_{\mathrm{w}} + S_{\mathrm{o}} + S_{\mathrm{g}} = 1$, the criterion $\partial k_{\mathrm{ro}}/\partial S_{\mathrm{o}} \geqslant 0$ is equivalent to $\partial k_{\mathrm{ro}}/\partial S_{\mathrm{g}} \leqslant 0$, which turns out to be more natural to show.

Proposition 2. *Assume $dk_{rog}/dS_g \leqslant 0$. Then for saturations in D, the Stone I model satisfies $\partial k_{ro}/\partial S_g \leqslant 0$ provided $\partial S_{\mathrm{om}}/\partial S_{\mathrm{g}} \geqslant -\frac{1}{2}$.*

Proof. The Stone I model is defined as

$$k_{\mathrm{ro}}(S_{\mathrm{w}}, S_{\mathrm{g}}) = k_{\mathrm{rocw}} S_{\mathrm{o}}^{*} \beta_{\mathrm{w}} \beta_{\mathrm{g}},$$

where

$$\beta_{\mathrm{w}} = \frac{k_{\mathrm{row}}(S_{\mathrm{w}})/k_{\mathrm{rocw}}}{1 - S_{\mathrm{w}}^{*}}, \quad \beta_{\mathrm{g}} = \frac{k_{\mathrm{rog}}(S_{\mathrm{g}})/k_{\mathrm{rocw}}}{1 - S_{\mathrm{g}}^{*}},$$

$k_{\mathrm{rocw}} = k_{\mathrm{row}}(S_{\mathrm{w}} = S_{\mathrm{wc}})$, and the normalized saturations are defined as

$$S_{\mathrm{w}}^{*} = \frac{S_{\mathrm{w}} - S_{\mathrm{wc}}}{1 - S_{\mathrm{wc}} - S_{\mathrm{om}}}, \quad S_{\mathrm{o}}^{*} = \frac{S_{\mathrm{o}} - S_{\mathrm{om}}}{1 - S_{\mathrm{wc}} - S_{\mathrm{om}}}, \quad S_{\mathrm{g}}^{*} = \frac{S_{\mathrm{g}}}{1 - S_{\mathrm{wc}} - S_{\mathrm{om}}}.$$

Combining all these relations, we see that $k_{\mathrm{ro}} = U(S_{\mathrm{w}}, S_{\mathrm{g}}, S_{\mathrm{om}})/V(S_{\mathrm{w}}, S_{\mathrm{g}}, S_{\mathrm{om}})$, where

$$U = (1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}})(1 - S_{\mathrm{wc}} - S_{\mathrm{om}}) k_{\mathrm{row}}(S_{\mathrm{w}}) k_{\mathrm{rog}}(S_{\mathrm{g}}),$$
$$V = (1 - S_{\mathrm{wc}} - S_{\mathrm{om}} - S_{\mathrm{g}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}}).$$

Remembering that $S_{\mathrm{om}} = S_{\mathrm{om}}(S_{\mathrm{w}}, S_{\mathrm{g}})$, we deduce that

$$\frac{\partial k_{ro}}{\partial S_{\mathrm{g}}} = \frac{1}{V^2}\left[\left(V\frac{\partial U}{\partial S_{\mathrm{g}}} - U\frac{\partial V}{\partial S_{\mathrm{g}}}\right) + \frac{\partial S_{\mathrm{om}}}{\partial S_{\mathrm{g}}}\left(V\frac{\partial U}{\partial S_{\mathrm{om}}} - U\frac{\partial V}{\partial S_{\mathrm{om}}}\right)\right] = \frac{1}{V^2}\left[R_1 + R_2 \cdot \frac{\partial S_{\mathrm{om}}}{\partial S_{\mathrm{g}}}\right],$$

so the sign of $\partial k_{\mathrm{ro}}/\partial S_{\mathrm{g}}$ is determined by the quantity within the square brackets. After some manipulation, we get

$$\begin{aligned} R_1 &= -(S_{\mathrm{w}} - S_{\mathrm{wc}})(1 - S_{\mathrm{wc}} - S_{\mathrm{om}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}}) k_{\mathrm{row}} k_{\mathrm{rog}} + (1 - S_{\mathrm{wc}} - S_{\mathrm{om}} - S_{\mathrm{g}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}}) \\ &\quad \times (1 - S_{\mathrm{wc}} - S_{\mathrm{om}}) \times (1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}}) k_{\mathrm{row}} k'_{\mathrm{rog}} \\ &\leqslant -(S_{\mathrm{w}} - S_{\mathrm{wc}})(1 - S_{\mathrm{wc}} - S_{\mathrm{om}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}}) k_{\mathrm{row}} k_{\mathrm{rog}} \leqslant 0, \end{aligned}$$

since $k'_{\mathrm{org}} \leqslant 0$. In addition, we get

$$R_2 = -S_{\mathrm{g}}(S_{\mathrm{w}} - S_{\mathrm{wc}})[(1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}}) + (1 - S_{\mathrm{wc}} - S_{\mathrm{om}})] k_{\mathrm{row}} k_{\mathrm{rog}} \leqslant 0.$$

Hence $R_1 + R_2 \cdot \frac{\partial S_{\mathrm{om}}}{\partial S_{\mathrm{g}}} \leqslant 0$ if either $\partial S_{\mathrm{om}}/\partial S_{\mathrm{g}} \geqslant 0$ or

$$\left|\frac{\partial S_{\mathrm{om}}}{\partial S_{\mathrm{g}}}\right| \leqslant \frac{(S_{\mathrm{w}} - S_{\mathrm{wc}})(1 - S_{\mathrm{wc}} - S_{\mathrm{om}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}})}{S_{\mathrm{g}}(S_{\mathrm{w}} - S_{\mathrm{wc}})[(1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}}) + (1 - S_{\mathrm{wc}} - S_{\mathrm{om}})]}. \tag{B.1}$$

But since $S_{\mathrm{g}} \leqslant 1 - S_{\mathrm{w}} - S_{\mathrm{om}}$ and $1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}} \leqslant 1 - S_{\mathrm{wc}} - S_{\mathrm{om}}$, we see that

$$\frac{(S_{\mathrm{w}} - S_{\mathrm{wc}})(1 - S_{\mathrm{wc}} - S_{\mathrm{om}})(1 - S_{\mathrm{w}} - S_{\mathrm{om}})}{S_{\mathrm{g}}(S_{\mathrm{w}} - S_{\mathrm{wc}})[(1 - S_{\mathrm{w}} - S_{\mathrm{om}} - S_{\mathrm{g}}) + (1 - S_{\mathrm{wc}} - S_{\mathrm{om}})]} \geqslant \frac{1}{2}.$$

Thus, in order to ensure that $\partial k_{ro}/\partial S_g \leqslant 0$, it is sufficient to require either $\partial S_{om}/\partial S_g \geqslant 0$ or $|\partial S_{om}/\partial S_g| \leqslant \frac{1}{2}$, which is equivalent to requiring $\partial S_{om}/\partial S_g \geqslant -\frac{1}{2}$. □

Note that if the Fayers and Matthews [8] model for S_{om} is used, we would have

$$\frac{\partial S_{om}}{\partial S_g} = -\frac{S_{orw} - S_{org}}{1 - S_{wc} - S_{org}},$$

so the condition in Proposition 2 would be satisfied as long as $S_{orw} - S_{org}$ is small, which is usually the case. In particular, the monotonicity condition is always satisfied whenever $S_{orw} = S_{org}$.

Proposition 3. *Assume that* $dk_{rg}/dS_g \geqslant 0$, $dk_{rog}/dS_g \leqslant 0$, *and that* k_{rw} *and* k_{row} *are convex functions of* S_w. *Then for all saturations in D, the Stone II model satisfies* $\partial k_{ro}/\partial S_g \leqslant 0$.

Proof. The Stone II model is defined as

$$k_{ro}(S_w, S_g) = k_{rocw}\left[\left(\frac{k_{row}}{k_{rocw}} + k_{rw}\right)\left(\frac{k_{rog}}{k_{rocw}} + k_{rg}\right) - \left(k_{rw} + k_{rg}\right)\right].$$

Differentiating with respect to S_g gives

$$\frac{\partial k_{ro}}{\partial S_g} = \left(\frac{k_{row}}{k_{rocw}} + k_{rw} - 1\right)k'_{rg} + \left(\frac{k_{row}}{k_{rocw}} + k_{rw}\right)\frac{k'_{rog}}{k_{rocw}}.$$

The second term is clearly non-positive because $k'_{rog} \leqslant 0$. To show that the first term is also non-positive, first note that $k'_{rg} \geqslant 0$. Next, define $g(S_w) = k_{rw} + k_{row}/k_{rocw}$. Then $g(S_{wc}) = g(1 - S_{orw}) = 1$. But since g is convex, it must be that $g(S_w) \leqslant 1$ for all $S_{wc} \leqslant S_w \leqslant 1 - S_{orw}$. So $g(S_w) - 1 \leqslant 0$, which implies the first term is non-positive as well. Hence, we have shown that $\partial k_{ro}/\partial S_g \leqslant 0$, as required. □

References

[1] John R. Appleyard, Ian M. Cheshire, The cascade method for accelerated convergence in implicit simulators, in: European Petroleum Conference, 1982, pp. 113–122.
[2] Khalid Aziz, Antonin Settari, Petroleum Reservoir Simulation, Applied Science Publishers, New York, 1979.
[3] R.P. Brent, Algorithms for Minimization without Derivatives, Prentice-Hall, Englewood Cliffs, NJ, 1973 (Chapter 3–4).
[4] Hui Cao. Development of Techniques for General Purpose Simulators, PhD thesis, Stanford University, Stanford, CA, June 2002.
[5] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, SPE Reservoir Eval. Eng. 4 (4) (2001) 308–317.
[6] K.H. Coats, A note on IMPES and some IMPES-based simulation models, SPEJ 5 (3) (2000) 245–251.
[7] E.F. D'Azevedo, P.A. Forsyth, W.-P. Tang, Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems, SIAM J. Matrix Anal. Appl. 13 (3) (1992) 944–961.
[8] F.J. Fayers, J.D. Matthews, Evaluation of normalized Stone's methods for estimating three-phase relative permeabilities, SPE J. 24 (1984) 224–232.
[9] Felix Kwok. Scalable Linear and Nonlinear Algorithms for Multiphase Flow in Porous Media, PhD thesis, Stanford University, Stanford, CA, December 2007.
[10] Aziz S. Odeh, Comparison of solutions to a three-dimensional black-oil reservoir simulation problem, JPT (1981) 13–25.
[11] Walter Rudin, Principles of Mathematical Analysis, third ed., McGraw-Hill, 1976.
[12] Youcef Saad, Martin H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[13] J.R. Wallis, R.P. Kendall, T.E. Little, Constrained residual acceleration of conjugate residual methods, in: Eighth SPE Symposium on Reservoir Simulation, 1985, pp. 415–428.