Two-stage Preconditions for Inexact Newton Methods in Multi-phase Reservoir Simulation

Hector Klie
Marcelo Rame
Mary Wheeler

CRPC-TR96660
October 1996

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Two-stage preconditioners for inexact Newton methods in multi-phase reservoir simulation [1]

Héctor Klíe [a,2], Marcelo Ramé [a,3] and Mary F. Wheeler [b,4]

[a] *Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA.*

[b] *Texas Institute for Computational and Applied Mathematics, University of Texas, Austin, Texas 78712, USA.*

Abstract

We discuss two-stage preconditioners for solving systems of coupled nonlinear partial differential equations in the modeling of underground multiphase flow phenomena. The linear systems arising from the discretization and Newton linearization are nonsymmetric and indefinite but coefficient blocks associated with a particular type of unknown possess properties that can be exploited to improve the conditioning of the coupled system. We show through theoretical discussion and numerical experiments that decoupling strategies combined with two-stage preconditioners are more effective to accelerate Krylov subspace methods such as GMRES and BiCGSTAB than standard ones which "blindly" precondition the entire coupled linear system. We also show a distributed memory parallel implementation of some of the iterative schemes proposed in this work.

1 Introduction

Nowadays, the idea of solving partial differential equations (PDE's) involving millions of unknowns is becoming plausible and attractive to the numerical analyst and the application programmer in science and engineering. In particular, the need for solving such large problems with several unknowns per gridblock has become one of the main challenges in the reservoir community. Therefore

[1] This work was partially supported by Intevep S.A., Los Teques, Edo. Miranda, Venezuela, ONR grant N00014-93-1-0158 and CRPC.

[2] klie@rice.edu

[3] marcelo@rice.edu

[4] mfw@ticam.utexas.edu

the conception of robust and efficient iterative solvers plays an important role in the oil industry research in connection to solving coupled sets of nonlinear equations as obtained by a fully implicit discretization of multi-phase models.

In this work we focus our attention on two-stage procedures (also known in the literature as nested or inner-outer procedures; see e.g., [2,13,17,20]). We address their use as preconditioners for the several large sparse linear systems arising from the cell-centered finite difference or, equivalently, lowest-order mixed finite element discretization (with an appropiate quadrature rule; see [32]) and the subsequent Newton linearization of the coupled algebraic system of nonlinear equations. These linear systems (i.e., instances of Newton equations) are nonsymmetric and highly indefinite. Not surprisingly, specific preconditioners for these type of problems are not frequent in the literature due in part to the complexity suggested by the contrasting physical behavior of the variables involved: pressures (elliptic or parabolic component) and saturations (hyperbolic or convection-dominated component.)

Despite the difficulty of these linear systems, there are certainly some "nice" properties associated to the coefficient blocks that affect each family of nodal unknowns. Under mild conditions regularly met at a modest time step size, each of these blocks are irreducible and diagonally dominant. Additionally, the strict diagonal dominance in some of these blocks leads to the M-matrix property. These algebraic properties can be exploited so that better conditioning can be achieved in the entire coupled system. Moreover, devices leading to this desirable situation help weaken the coupling between the nonlinear partial differential equations represented by the off-diagonal blocks. We call these devices decoupling operators and use them as a preprocessing step to enhance the effectiveness of two-stage preconditioners. These preconditioners are aimed at adding efficiency and robustness to two well known Krylov subspace iterative methods: GMRES and BiCGSTAB. Results of this work are applicable to hybrid Krylov-secant methods developed by the authors [25] and also to higher order Krylov-based inexact Newton methods [23].

This paper is organized as follows. Section 2 covers the model equations, their discretization and linearization by the Newton method. In Section 3, we analyze the structure of the Jacobian linear system. Section 4 focuses the discussion on two different decoupling operators and their ability to cluster the eigenvalues of the original coupled system. Section 5 is devoted to discussing the family of two-stage procedures and to describing those preconditioners that we consider most appropriate in our context. Technical discussion is supported by experiments in Section 6. Conclusions are given in Section 7.

2 Description of the Problem

The paper concentrates the analysis on the equations for two-phase black-oil simulation which constitute the simplest way to realistically model multi-phase flow and transport in porous underground formations. Extensions to more unknowns per gridblock are readily evident.

2.1 Differential Equations

The basic equations for two-phase black-oil simulation consist of conservation equations for a wetting (i.e., water) and a non-wetting (i.e., oil) phase, denoted by subscripts w and n, respectively. A more thorough description of the model can be found in [3]. The mass conservation of each phase is given by

$$\frac{\partial(\phi\rho_n S_n)}{\partial t} + \nabla\cdot(\rho_n \mathbf{u}_n) = q_n, \tag{1}$$

$$\frac{\partial(\phi\rho_w S_w)}{\partial t} + \nabla\cdot(\rho_w \mathbf{u}_w) = q_w, \tag{2}$$

where ρ_l is the density, ϕ is the porosity, S_l is the saturation, t is time, q_l is the source term with denotes the production/injection rates at reservoir conditions, and $\mathbf{u}_l$ is the phase Darcy velocity which is expressed as

$$\mathbf{u}_l = -\frac{\mathrm{K}k_{rl}}{\mu_l}\left(\nabla P_l - \rho_l g \nabla Z\right),$$

where K is the absolute permeability tensor, k_{rl} is the relative permeability, μ_l is the viscosity, P_l is the pressure, g is the gravity and Z is the depth. The subscript l can be either w for the wetting or n for the non-wetting phase. These equations are coupled through the following extra relations:

- Wetting and non-wetting saturations add up to one: $S_w + S_n = 1$.
- Capillary pressure: $P_c\left(S_w\right) = P_n - P_w$.
- Relative permeabilities depend on both location and saturation.

The model also allows for slight compressibility of both phases, i.e., $\rho_l\left(P_l\right) = \rho_{0l}e^{c_l P_l}$, where ρ_{0l} and c_l are given physical constants. Absolute permeability, porosity, viscosity, capillary pressure and depth depend only on location.

The simulator used in the experiments presented in this work can accommodate problems from both the petroleum and the environmental engineering

disciplines for it can specify general boundary conditions given by

$$\sigma \mathbf{u}_w \cdot \vec{n} + \nu P_w = h_w \text{ and } \sigma \mathbf{u}_n \cdot \vec{n} + \nu P_n = h_n,$$

where σ and ν are spatially varying coefficients, $\vec{n}$ is the outward, unit, normal vector and h_l is a spatially varying function. Initially, P_n and S_w are specified. A gravity equilibrium condition is then used to solve for an initial value of S_n. The primary unknowns in our simulator are P_n and $c_n = \rho_n S_n$, i.e., the nonwetting phase *concentration* (or, perhaps, more properly *mass per unit pore volume*). All other variables can then be computed explicitly based on these two.

In the case of slight compressibility, our system in the primary unknowns includes one nonlinear parabolic equation in terms of pressure and one nonlinear convection-diffusion equation in terms of concentration [3]. In this model, there are weak nonlinearities related to those variables that depend upon pressures of one phase (e.g., densities) and their effect depends on the degree of pressure change. In contrast, strong nonlinearities are present in variables that depend on concentrations such as relative permeability and capillary pressure. The pressure equation degenerates into an elliptic equation when both phases are incompressible (i.e., $c_n = c_w = 0$). On the other hand, the diffusive term in the latter equation vanishes in the absence of capillary pressure, giving rise to a first order quasi-linear hyperbolic equation.

2.2 Discretization

In the context of the two-phase problem being discussed in this work, both pressure and concentration unknowns (degrees of freedom) occupy the centers of the discretization blocks and velocities are approximated on the edges or faces of the discretization blocks. The components of the flow coefficients or mass mobilities, λ_l $(l = n, w)$ between two grid elements are defined as follows

$$\lambda^{T+1}_{l,i+1/2,jk} = \left(\frac{\rho_l k_l}{\mu_l}\right)^{T+1}_{i+1/2,j,k} K_{i+1/2,jk},$$

where the superscript $T+1$ denotes the $(T+1)$-th approximation of the Newton iterates to a value at the $(n+1)$-th time level; the subscripts i, j and k indicate the grid block location. The bracketed factor on the right hand side is approximated through upstream weighting and the permeability is weighted harmonically in the flow direction to account for variations in grid block sizes.

Discretization of the model equations (1)-(2) is performed by block-centered finite differences (or, equivalently, by lowest-order mixed finite elements) obey-

ing, for a diagonal premeability tensor, a seven point stencil for pressures and concentrations of both phases, thus giving rise to 28 coefficients associated with an internal node. This results in the nonlinear algebraic system

$$
\begin{aligned}
V_{ijk}((\rho_l S_l)_{ijk}^{T+1} - (\rho_l S_l)_{ijk}^{n}) &= \Delta t^n \Delta x_i \Delta y_j \Delta z_k q_{l,ijk}^{T+1} \\
&+ \Delta t^n \Delta y_j \Delta z_k \Bigg\{ \lambda_{l,i+\frac{1}{2}} \frac{P_{l,i+1,jk} - P_{l,ijk}}{\Delta x_{i+1/2}} - \\
&- [\lambda_{l,i+\frac{1}{2}} \rho_l g] \frac{Z_{i+1,jk} - Z_{ijk}}{\Delta x_{i+1/2}} - \\
&- \lambda_{l,i-\frac{1}{2}} \frac{P_{l,ijk} - P_{n,i-1,jk}}{\Delta x_{i-1/2}} + \\
&+ [\lambda_{l,i-\frac{1}{2}} \rho_l g] \frac{Z_{ijk} - Z_{i-1,jk}}{\Delta x_{i-1/2}} \Bigg\}_{jk}^{T+1} \\
&+ \text{similar terms for the y and z directions,}
\end{aligned}
\tag{3}
$$

where $\Delta x_{i+1/2} = (x_{i+1} - x_i)/2$, i.e., the cell midpoint along the x direction and $V_{ijk} = \Delta x_i \Delta y_j \Delta z_k \phi_{ijk}$, i.e., the pore volume. In a similar way, $\Delta y_{i+1/2}$ and $\Delta z_{i+1/2}$ are defined. Dawson *et al.* [12] extended this to a 19-point stencil in space and use a full permeability tensor to handle underground heterogeneities. Numerical experiments in Section 6 are done using this extension.

The extra relations mentioned in the previous subsection and their partial differentiation with respect the primary unknowns are used in obtaining the Newton linearization of the nonlinear conservation equations. Small compressibility allows for some simplifications, without affecting the validity of the numerical approximation. The above procedure follows the description by Wheeler and Smith [33] on developing a parallel black-oil simulator. Further insights about discretization of these equations can be found in [3].

2.3 Newton framework

The fully implicit formulation of the two-phase black-oil equations leads to the solution of (3) for each time step, which can be abstractly represented by $F(u) = 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$. Here, the vector u represents pressure and concentration unknowns of the non-wetting phase.

The composition of Newton with a Krylov iterative solver with a criterion for defining linear tolerances dynamically, and a line-search backtracking strategy

[14] is the basis of our inexact Newton algorithm and is described as follows:

Algorithm 1 (Inexact Newton method)

1. Let $u^{(0)}$ be an initial guess.
2. For $k = 0, 1, 2, \ldots$ until convergence do
2.1 Choose $\eta^{(k)} \in [0, 1)$.
2.2 Using some Krylov iterative method, compute a vector $s^{(k)}$ satisfying

$$J^{(k)} s^{(k)} = -F^{(k)} + r^{(k)}, \tag{4}$$

with $\frac{\|r^{(k)}\|}{\|F(u^{(k)})\|} \leq \eta^{(k)}$.
2.3 Set $u^{(k+1)} = u^{(k)} + \lambda^{(k)} s^{(k)}$.

The residual solution, $r^{(k)}$, represents the amount by which the solution, $s^{(k)}$, given by the Krylov iterative solver fails to satisfy the *Newton equation* (or *Jacobian equation*), i.e., $J^{(k)} s^{(k)} = -F^{(k)}$. The step length $\lambda^{(k)}$ is computed by a line-search backtracking method to ensure a decrease of $f(u) = \frac{1}{2} F(u)^t F(u)$. The solution $s^{(k)}$ of (4) should be a descent direction for $f(u^{(k)})$ [14]. In practice, the final residual given by the iterative linear solver is acceptable whenever the *Dembo-Eisenstat-Steihaug condition* is met [13], i.e.,

$$\left\| r_m^{(k)} \right\| = \left\| F^{(k)} + J^{(k)} s_m^{(k)} \right\| \leq \eta^{(k)} \left\| F^{(k)} \right\|, \qquad 0 < \eta^{(k)} < \eta_{max} < 1, \tag{5}$$

where m indicates the number of linear iterations employed. The linear tolerance $\eta^{(k)}$ is known as the *forcing term* of (5). Heuristics to select linear tolerances or forcing terms according to the progress of the inexact Newton method are given in [15,16]. The main goal here is to prevent oversolving of the equation whenever there is a considerable disagreement between the nonlinear function F and its local linear model. Practical experiences on this ideas in the context of reservoir simulation are reported in [12]. Finally, full description and several pointers into the literature of Krylov subspace iterative methods can be found in [5]. Their use as linear solvers within Newton's method are detailed for instance, in [10].

3 The algebraic coupled linear system framework

We now provide general description of the linear systems (i.e., Newton equation) arising in step 2.2 of Algorithm 1. We identify properties associated with the blocks of the partitioned system and establish some moderate assumptions to facilitate the analysis and the development of the procedures on which the preconditioners are based. These assumptions are not intended

to give a definitive characterization of real life simulation matrices but are met when the time step is short enough to ensure convergence of the Newton method itself and, therefore, provide a framework for evaluating the last advances in preconditioning coupled linear systems in reservoir engineering.

3.1 Structure of Resulting Linear System

Each linear system associated with the two phase model depicted in (1)-(2) can be partitioned in the following 2×2 block form

$$Jx = f \Leftrightarrow \begin{pmatrix} J_{pp} & J_{pc} \\ J_{cp} & J_{cc} \end{pmatrix} \begin{pmatrix} p \\ c \end{pmatrix} = - \begin{pmatrix} f_n \\ f_w \end{pmatrix}. \tag{6}$$

Each block $J_{i,j}, i, j = s, p$ is of size $nb \times nb$, where nb, is the number of grid blocks and $f_n\,(f_w)$, is the residual vector corresponding to the non-wetting (wetting) phase coefficients.

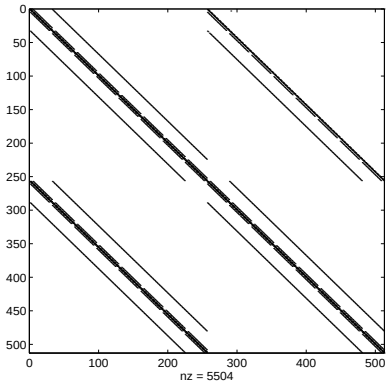

Fig. 1. Matrix structure of linear systems in the Newton iteration.

Each group of unknowns is numbered in a sequential lexicografic fashion: the pressure unknowns are numbered from one through the total number of grid blocks (nb) and the concentrations are numbered from $nb + 1$ through $2nb$.

The block J_{pp} has the structure of a purely elliptic problem in the non-wetting phase pressures. The block J_{pc} of the Jacobian matrix has a structure similar to that of a discretized first-order hyperbolic problem in the non-wetting phase concentrations. J_{cp} has the coefficients of a convection-free parabolic problem in the non-wetting phase pressure and, finally, J_{cc} represents a parabolic (convective-diffusive) problem in the oil concentrations. The sparsity pattern of a given Jacobian matrix is shown in Figure 1. We can observe the effect of the upstream weighting within the block J_{pc} with no superdiagonal coefficients, which are added to the main diagonal of that block.

3.2 An algebraic analysis of the coupled Jacobian matrix

The presence of slight compressibility ensures invertibility of the Jacobian matrix [3]. In general, in system (6), the block coefficients J_{pp}, J_{pc} and J_{cp} share the following properties (see e.g., [3] for further physical insights and [2] for mathematical definitions and related theoretical results):

- Diagonal dominance,
- Positive diagonal entries and negative off-diagonal entries (i.e., they are Z-matrices), and
- Irreducibility.

Strict diagonal dominance in all rows is only present in J_{pc} and J_{cp} as result of compressibility and pore volume term contributions to the main diagonal of these blocks. In consequence, these blocks are nonsingular, positive stable and M-matrices. Strict diagonal dominance for some of the rows of J_{pp} can be achieved by the contribution of bottom hole pressures specified as part of the boundary conditions, making this block irreducible and diagonally dominant. In addition, under small changes of formation volume factors [5] and flow rates between adjacent grid blocks both blocks J_{pp} and J_{cp} are nearly symmetric.

The concentration coefficient block $-J_{cc}$ presents algebraic properties similar to the other blocks. It has a convection-diffusion flavor characterized by capillary pressure derivative terms (the diffusive part) and wetting phase relative permeability derivative terms (the convective part). The diffusive part becomes dominant over the convective part when capillary pressure gradients are higher than relative permeability gradients of the wetting phase. This is likely to occur at the beginning and end of the of the simulation when the capillary pressure derivative with respect to wetting phase concentration is highest.

Desirable diagonal dominance in $-J_{cc}$ can indeed be achieved by shortening the time step. We have observed that the conditioning of this block has an immediate incidence on the conditioning of the whole system. Moreover, loss of diagonal dominance of this block not only affects negatively the linear solver's convergence rate but also compromises the Newton convergence even with a fairly accurate solution of the linear system. Hence, it appears that this block is crucial in determining the conditioning of the entire system. The reader can verify the resemblance between the spectrum of J_{cc} and that of the Jacobian J through inspection of Figure 2 and Figure 3.

[5] Formation volume factors of each phase are defined as the ratio of the volume occupied by the phase at reservoir conditions to the volume occupied at *stock-tank* or atmospheric conditions.

We should stress that the "degree" of diagonal dominance is proportional to the pore volume of the grid blocks and inversely proportional to the time step size (except for J_{pp}, whose diagonal dominance is independent of Δt). On the other hand, definition of bottom hole pressures as part of the boundary conditions enhances the diagonal dominance of the blocks, whereas specified rates in the source wells do not affect the diagonal dominance in any way.

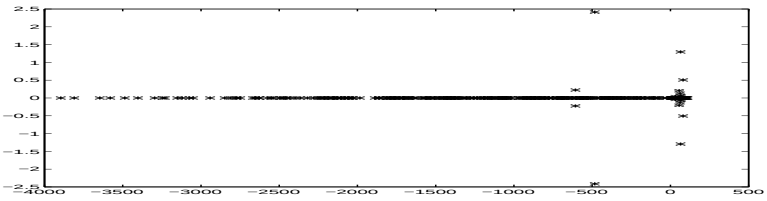

Fig. 2. Spectrum of the sample Jacobian matrix to be used throughout the discussion on two-stage preconditioners.

In this work, we assume the blocks J_{pp} and $-J_{cc}$ are irreducibly diagonally dominant and the blocks J_{pc} and J_{cp} are diagonally dominant. With the minus sign in front of J_{cc} all blocks are Z-matrices with positive diagonal entries. These conditions do not guarantee nice properties on the whole matrix J. Moreover, the Jacobian matrix is highly non-symmetric and indefinite in principle. This is the main argument in favor of decoupling strategies to generate preconditioners for (6) since better convergence behavior can be obtained by exploiting the algebraic properties of the Jacobian blocks.

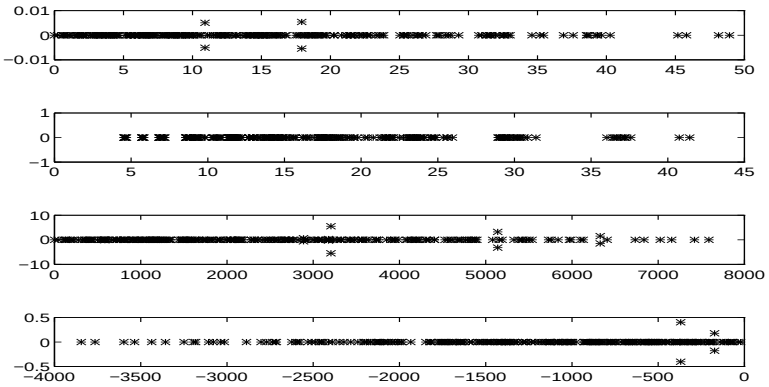

Fig. 3. Spectra of the blocks composing the sample Jacobian matrix. From top to bottom, they correspond to J_{pp}, J_{pc}, J_{cp} and J_{cc}.

In the next section, we progressively illustrate our analysis by looking at spectrum changes of a typical Jacobian matrix after applying different operators, resulting from a small scale reservoir simulation (i.e., a problem with grid size of $8 \times 8 \times 4$) where the blocks J_{pp}, J_{pc}, J_{cp} and $-J_{cc}$ are positive stable, as clearly depicted in Figure 3. Figure 2 shows that the matrix is indeed highly indefinite. Note that although the eigenvalues are largely spread along the negative real axis the Sylvester's law of inertia ensures that there are at least nb (i.e., half of the total) eigenvalues with positive real part.

4 Decoupling operators

The idea of decoupling operators has not only been barely stated in the general literature but also has been treated only in passing for reservoir simulation linear solvers. Somehow their potential as effective preconditioners for coupled systems has been underestimated or overlooked, perhaps due to the assumption that pressure based preconditioners account for all the dominant effects in the system. Unfortunately, this is no longer true under large changes in concentrations likely occurring at high flow rates or at larger time steps.

Bank *et al.* [4], with their *alternate-block factorization (ABF) method*, propose a simple way to weaken the coupling of drift-diffusion equations that occur in semiconductor device modeling. They use greatly simplifying assumptions, however, to analyze the decoupling process for preconditioning linear systems. Recent experiences with Bank *et al.* decoupling operator are reported in [19]. Their work has value in the context of multi-phase flow since their decoupling operator leads to a significant clustering of eigenvalues associated with Jacobian matrices occurring during the simulation process. Moreover, in very rare cases (detected only after extensive experimentation with random matrices whose blocks obey our assumptions for the blocks of J), the resulting decoupled system fails to have all eigenvalues lying on the right half of the complex plane. This suggests the convenience of employing such decoupling operators for removing a high degree of indefiniteness in the original linear system.

4.1 Block decoupling

Consider the Jacobian system shown in (6) and let us define

$$D = \begin{pmatrix} D_{pp} & D_{pc} \\ D_{cp} & D_{cc} \end{pmatrix} = \begin{pmatrix} diag(J_{pp}) & diag(J_{pc}) \\ diag(J_{cp}) & diag(J_{cc}) \end{pmatrix}, \tag{7}$$

that is, a matrix of 2×2 blocks each of them containing the main diagonal of the corresponding block of J. It clearly follows that

$$\begin{aligned} J^D \equiv D^{-1}J &= \begin{pmatrix} \Delta^{-1} & 0 \\ 0 & \Delta^{-1} \end{pmatrix} \begin{pmatrix} D_{cc}J_{pp} - D_{pc}J_{cp} & D_{cc}J_{pc} - D_{pc}J_{cc} \\ D_{pp}J_{cp} - D_{cp}J_{pp} & D_{pp}J_{cc} - D_{cp}J_{pc} \end{pmatrix} \\ &\equiv \begin{pmatrix} J^D_{pp} & J^D_{pc} \\ J^D_{cp} & J^D_{cc} \end{pmatrix}, \end{aligned} \tag{8}$$

where $\Delta \equiv D_{pp}D_{cc} - D_{pc}D_{cp}$, and the superscript D have been introduced for later notational convenience. All main diagonal entries of the main diagonal blocks are equal to one and those of the off-diagonal blocks are all equal to zero. In fact, we can expect that the degree of coupling of the off-diagonal blocks of J has been reduced to some extent. Bank *et. al.* [4] observe that this operation weakens the coupling between the partial differential equations. Note that the operation is simple to carry out and may not imply alterations to the underlying data structure holding the coefficients (e.g., diagonal matrix storage). In this case, five diagonals of length nb are enough to go back and forth between the original system J and the partially decoupled system J^D. In physical terms, the decoupling operator tends to approximate pressure coefficients as if concentration derivatives were neglected in the transmissibility computation. Hence, this is like "time-lagging" or evaluating some transmissibilities explicitly. We prefer the form $D^{-1}J$ over JD^{-1} since the latter may spoil the inherent diagonal dominance of J.

The ABF decoupling operator admits another representation, by associating smaller matrix blocks with unknowns sharing a grid block. This means to permute rows and columns of J, i.e., to number every pressure unknown followed by the concentration unknown at the same grid block (interleaved numbering). Let P be the matrix that performs such permutation and define

$$\widetilde{J} = PJP^t = \begin{pmatrix} \widetilde{J}_{1,1} & \widetilde{J}_{1,2} & \cdots & \widetilde{J}_{1,nb} \\ \widetilde{J}_{2,1} & \widetilde{J}_{2,2} & & \widetilde{J}_{2,nb} \\ \vdots & & \ddots & \vdots \\ \widetilde{J}_{nb,1} & \widetilde{J}_{nb,2} & \cdots & \widetilde{J}_{nb,nb} \end{pmatrix},$$

where

$$\widetilde{J}_{i,j} = \begin{pmatrix} (J_{pp})_{i,j} & (J_{pc})_{i,j} \\ (J_{cp})_{i,j} & (J_{cc})_{i,j} \end{pmatrix},$$

is the 2×2 matrix representing the coupling between unknowns sharing a grid block. It clearly follows for an invertible D that $\widetilde{D}^{-1} = PD^{-1}P^t$. Hence, $\widetilde{D}^{-1}$ is a block diagonal matrix whose blocks are the inverses of the Jacobian blocks

associated to a local problem at each grid block. That is,

$$\widetilde{D}^{-1} = \begin{pmatrix} \widetilde{J}_{1,1}^{-1} & 0 & \cdots & 0 \\ 0 & \widetilde{J}_{2,2}^{-1} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \widetilde{J}_{nb,nb}^{-1} \end{pmatrix}. \tag{9}$$

To follow the underlying notation, let us define the alternate decoupled system as $\widetilde{J}^D \equiv \widetilde{D}^{-1}\widetilde{J} = PD^{-1}JP^t$. This idea appears rather natural. In fact, Behie and Vinsome [7] comment about the possibility of decoupling more equations in their method but only with respect pressure coefficients. They did not foresee the positive effect, as we shall note below, that a full decoupling of the grid block has in conditioning the system.

The core of the combinative approach is the effective solution of pressure-based systems, so there is no need to go beyond in the decoupling process as expressed in (9). The coefficients introducing the coupling with pressures within the grid block are zeroed out so that corresponding coefficients at neighboring grid blocks are expected to become small. To be more precise, let

$$\widetilde{W}_p = \begin{pmatrix} (\widetilde{W}_p)_1 & 0 & \cdots & 0 \\ 0 & (\widetilde{W}_p)_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & (\widetilde{W}_p)_{nb} \end{pmatrix}, \tag{10}$$

where $(\widetilde{W}_p)_i = I_{nu\times nu} - e_1 e_1^t + (e_1^t J_{ii} e_1)\, e_1 e_1^t \widetilde{J}_{ii}^{-1}$, and $e_1 = (1,0)^t$. Therefore, the operator $\widetilde{W}_p$ (introduced by Wallis in his IMPES two-stage preconditioner [31]) is a block diagonal matrix that removes the coupling in each 2×2 diagonal block with respect to the pressure unknown, i.e., it readily follows that

$$(\widetilde{W}_p)_i \widetilde{J}_{ii} = \widetilde{J}_{ii} - e_1 e_1^t \widetilde{J}_{ii} + \left(e_1^t \widetilde{J}_{ii} e_1\right) e_1 e_1^t = \begin{pmatrix} (J_{pp})_{i,j} & 0 \\ (J_{cp})_{i,j} & (J_{cc})_{i,j} \end{pmatrix}.$$

We could also define an operator $\widetilde{W}_c$ with the canonical vector $e_2 = (0,1)^t$. The consecutive counterpart, W_p, of the alternate operator $\widetilde{W}_p$ is given by

$$
\begin{aligned}
J^{W_p} \equiv W_p J &= \begin{pmatrix} \Delta^{-1} D_{pp} & 0 \\ 0 & I_{nb \times nb} \end{pmatrix} \begin{pmatrix} D_{cc} J_{pp} - D_{pc} J_{cp} & D_{cc} J_{pc} - D_{pc} J_{cc} \\ J_{cp} & J_{cc} \end{pmatrix} \\
&\equiv \begin{pmatrix} J_{pp}^{W_p} & J_{pc}^{W_p} \\ J_{cp}^{W_p} & J_{cc}^{W_p} \end{pmatrix}.
\end{aligned}
\tag{11}
$$

Clearly, the lower blocks are unmodified as well as the main diagonal of the resulting pressure block (i.e., $J_{cp}^{W_p} \equiv J_{cp}$ and $J_{cc}^{W_p} \equiv J_{cc}$.)

In order to reduce the already decoupled system to one involving a particular set of coefficients, say, those associated to pressure unknowns, the operator $R_p^t \in \mathbb{R}^{nb \times 2nb}$ is defined by

$$
(R_p^t)_{ij} = \begin{cases} 1 \qquad \text{if} \qquad i = k \text{ and } j = 1 + 2\,(k-1)\,, \\ 0 \text{ otherwise} \end{cases}
$$

for $k = 1, 2, \ldots, nb$. In this lexicographic alternate unknown numbering, we could also define $j = 2+2\,(k-1)$ for R_c^t to obtain the corresponding concentration coefficients. This presentation can be easily extended to more unknowns sharing a given grid point (e.g., three phases and multi-component systems.)

4.2 Properties of the partially decoupled blocks

In general, it is difficult (and in fact, an open problem in many related fields [1,4,9,18]) to characterize properties for the coupled Jacobian matrix and even more so if it has been affected by some of the operators described above. This is one of the reasons that theory concerning existence and applicability of different linear solvers or preconditioners is based on some specific assumptions on the matrix J. In our case, there is no theoretical result that determines when J is positive stable and moreover, when the symmetric part of J could have only positive eigenvalues although the matrix has some blocks that are M-matrices and present diagonal dominance.

In the applications of iterative solvers it is fundamental to have an idea of the spectrum of the operators on which they are applied, for if all eigenvalues fall on the right half of the complex plane, theoretical convergence of the iterative method is ensured. Also important is to detect a possible clustering of

the eigenvalues since this may increase the rate of convergence. Two immediate results can be shown for the individual diagonal blocks of the partially decoupled Jacobian through the action of D, as indicated in (8).

Theorem 1 *Let J_{pp} and $-J_{cc}$ be diagonally irreducibly Z-matrices and let J_{pc} and J_{cp} be M-matrices in $\mathbb{R}^{nb\times nb}$, then J^D_{pp} and J^D_{cc} are M-matrices.*

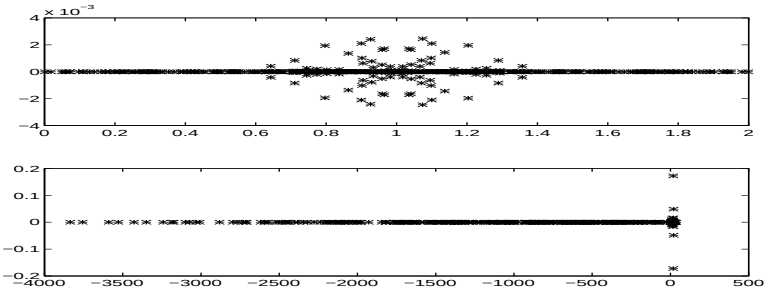

Fig. 4. Spectra of the partially decoupled forms of the sample Jacobian matrix. The one above corresponds to $D^{-1}J$, and the one below to $\widetilde{W}\widetilde{J}$ (or equivalently, WJ.)

The straightforward proof can be found in [23]. It can also be shown that the diagonal blocks J^D_{pp} and J^D_{cc} are positive stable [2, Theorem 6.12, page 211].

The effect of these properties can be seen in Figure 4 which shows the spectrum of the resulting Jacobian matrix after applying the decoupling operators D^{-1} and $\widetilde{W}$. The Jacobian spectrum has been significantly compressed and shifted to the right half of the complex plane by the action of D^{-1}. Several similar experiments indicate that breaking the coupling between unknowns is more effective at this than trying to preserve some desirable properties of the individual blocks, as intended by the design of W (see the great resemblance between the spectra of WJ and J).

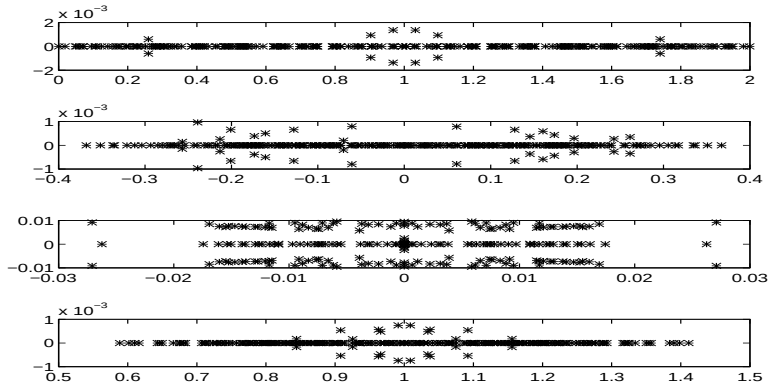

Fig. 5. Spectra of each block after decoupling with D. From top to bottom, they correspond to the (1,1), (1,2), (2,1) and (2,2) blocks

5 Two-stage preconditioners

5.1 A brief background

Appearence of decoupled preconditioners in reservoir simulation dates back to the *combinative* preconditioner of Behie and Vinsome [7], based on the solution of a reduced pressure system. A modification seeking to include global information was later proposed by Behie and Forsyth [6] and Wallis [30,31]. On the other hand, the concatenation or combination of inexact preconditioning stages has been proposed for general symmetric and non-symmetric problems [30] but specially in the context of domain decomposition [8,22] for flow in porous media. These efforts, however, do not address the topic of specialized preconditioners for coupled equations.

The use of two-stage methods is not new (see e.g., [26] and references cited there). They are also known as inner-outer or inexact iterations [13,20]. In the context of preconditioning they have been referred to as nonlinear, variable or inner-outer preconditioners [2,27]. They have been also subject of study in parallel computing settings (e.g. see [11] and further references therein.) However, in the context of large-scale systems of coupled equations they strangely seem to have been overlooked. The renewed interest in using two-stage methods is due to recent developments in Krylov-subspace methods that make the solution of the large inner linear systems affordable. For example, the Uzawa algorithm has been around for more than 35 years and it was only recently that some researchers formalized its inexact version (e.g., [17]). In same fashion, intensive work has been recently devoted to extending current non-symmetric iterative solvers to accommodate the inexactness or variability of the preconditioner from iteration to iteration; e.g. [2,27,29].

5.2 Combinative two-stage preconditioner

Consider the two-stage preconditioner M expressed as the solution of the preconditioned residual equation $M_p v = r$. Also, denote $\widetilde{J}^{W_p} \equiv \widetilde{W}_p \widetilde{J}$. Then the action of the preconditioner M_p is described by the following steps,

Algorithm 2 (Two-stage Combinative)

1. Solve $\left(R_p^t \widetilde{J}^{W_p} R_p\right) p = R_p^t \widetilde{W}_p r$ and denote its solution by $\hat{p}$.
2. Obtain expanded solution $p = R_p \hat{p}$.
3. Compute new residual $\hat{r} = r - \widetilde{J} p$.
4. Precondition and correct $v = \hat{M}^{-1} \hat{r} + p$.

The action of the whole preconditioner can be compactly written as

$$v = M_p^{-1} r = \hat{M}^{-1} \left[I - \left(\tilde{J} - \hat{M} \right) R_p \left(R_p^t \tilde{J}^{W_p} R_p \right)^{-1} R_p^t \widetilde{W}_p \right] r. \tag{12}$$

The preconditioner $\hat{M}$ is to be preferably recomputed for each Newton iteration, so it should be easily factored. Solving $\left(R_p^t \tilde{J}^{W_p} R_p \right) p = R_p^t \widetilde{W}_p r$ iteratively gives rise to a nested procedure. Note that M_p is an exact left inverse of $\tilde{J}$ on the subspace spanned by the columns of R_p. That is, $\left(M_p^{-1} \tilde{J} \right) R_p = R_p$.

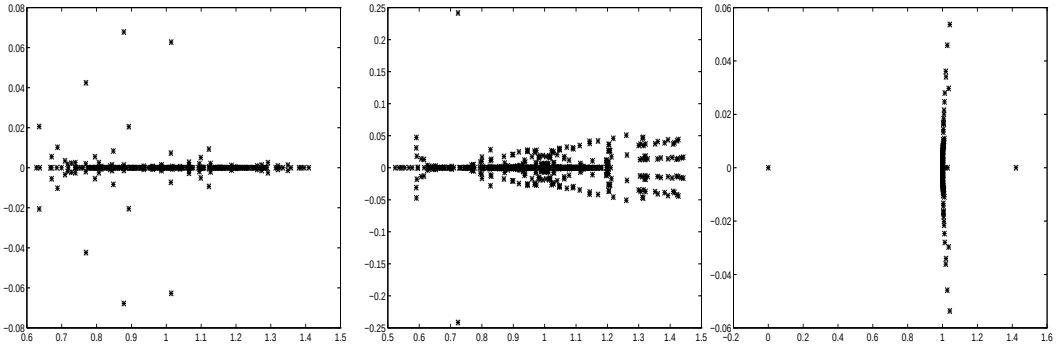

Fig. 6. Spectra of the Jacobian preconditioned on the right side by the exact version of the 2SComb (LEFT), by the 2SAdd (CENTER) and by the 2SMult preconditioners (RIGHT).

This is the preconditioner as stated by Wallis [31]. In contrast to the combinative method of Behie and Vinsome [7], he proposes to solve the pressure system iteratively and formalizes the form of the operators $\widetilde{W}_p$ and R_p. Although Wallis refers to the preconditioner as two-step IMPES preconditioner, we consider the term two-stage combinative preconditioner (2SComb) more appropriate, according to a more accepted terminology for convergent nested inexact procedures and to the former designation employed by Behie and Vinsome. Figure 6 (left graph) shows the spectrum of the operator for an exact solution of the pressure system, where $\hat{M}$ is the tridiagonal part of $\tilde{J}$.

5.3 Additive and multiplicative extensions

With the use of $\widetilde{D}^{-1}$ and incorporating the solution to a reduced concentration system (in addition to that to the reduced pressure system) we can improve the quality of the previous preconditioner. We propose to accomplish this additively or multiplicatively. The preconditioned residual $v = M_{\text{add}}^{-1} r$ in the

additive combinative two-stage scheme (2SAdd) is obtained by

Algorithm 3 (Two-stage Additive)

1. Solve $\left(R_p^t \widetilde{J}^D R_p\right) p = R_p^t \widetilde{D}^{-1} r$ and denote its solution by $\hat{p}$.
1. Solve $\left(R_c^t \widetilde{J}^D R_c\right) c = R_s^t \widetilde{D}^{-1} r$ and denote its solution by $\hat{c}$.
2. Obtain expanded solutions $p = R_p \hat{p}$ and $c = R_c \hat{c}$.
3. Add both approximate solutions $y = p + c$.
4. Compute new residual $\hat{r} = r - \widetilde{J} y$.
5. Precondition and correct $v = \hat{M}^{-1} \hat{r} + y$.

Instead, the multiplicative combinative two-stage preconditioner (2SMult) proposes the sequential treatment of the partially preconditioned residuals to obtain $v = M_{\text{mult}}^{-1} r$. In algorithmic terms it is given by

Algorithm 4 (Two-stage Multiplicative)

1. Solve $\left(R_p^t \widetilde{J}^D R_p\right) p = R_p^t \widetilde{D}^{-1} r$ and denote its solution by $\hat{p}$.
2. Obtain expanded solutions $p = R_p \hat{p}$.
3. Construct new residuals $\hat{r} = r - \widetilde{J} p$.
4. Solve $\left(R_c^t \widetilde{J}^D R_c\right) c = R_c^t \widetilde{D}^{-1} \hat{r}$ and denote its solution by $\hat{s}$.
5. Obtain expanded solutions $s = R_c \hat{c}$.
6. Compute new residual $w = r - \widetilde{J}(s + p)$.
7. Precondition and correct $v = \hat{M}^{-1} w + s + p$.

Assuming that both reduced pressures and concentrations are solved exactly and introducing the notation $t_l \equiv R_l \left(R_l^t \widetilde{J}^D R_l\right)^{-1} R_l^t \widetilde{D}^{-1}$, for $l = p, s$, the action of these preconditioners can be characterized by

$$v = M_{\text{add}}^{-1} r = \hat{M}^{-1} \left[I - \left(\widetilde{J} - \hat{M}\right)(t_p + t_c)\right] r, \tag{13}$$

and

$$v = M_{\text{mult}}^{-1} r = \hat{M}^{-1} \left[I - \left(\widetilde{J} - \hat{M}\right)\left(t_p + t_c - t_c \widetilde{J} t_p\right)\right] r. \tag{14}$$

The difference between the two preconditioners resides in the inclusion of the cross term $t_c \widetilde{J} t_p$ resulting from the computation of a new residual in Step 6 of Algorithm 4. Preliminary computational experiences with these preconditioners were presented in [24]. Figure 6 (center and right graphs) shows their effectiveness in clustering the spectrum around the point $(1, 0)$ of the complex plane. Note that the multiplicative two-stage preconditioner produces the major clustering of the real parts of the eigenvalues around unity among the three, although the resulting system has a negative eigenvalue.

5.4 Consecutive block two-stage preconditioners

In the same way that decoupling operators have interpretation in alternate block form, we can express the preconditioners described above in consecutive block form. However, in this opportunity we present them in a simpler form, given that the decoupling operator does a "good" job of clustering the spectrum of the original coupled system. In other words, we apply the block versions directly to J^D and omit the correcting step via $\hat{M}$ used in the combinative preconditioner and its corresponding additive and multiplicative extensions, since the overhead introduced by this operation is difficult to justify considering its limited preconditioning effectivity. For the ease of presentation we consider the factored form of the block-partitioned system (8),

$$J^D = \begin{pmatrix} I_{nb\times nb} & J^D_{pc}\left(J^D_{cc}\right)^{-1} \\ 0 & I_{nb\times nb} \end{pmatrix} \begin{pmatrix} S^D & 0 \\ 0 & J^D_{cc} \end{pmatrix} \begin{pmatrix} I_{nb\times nb} & 0 \\ \left(J^D_{cc}\right)^{-1} J^D_{cp} & I_{nb\times nb} \end{pmatrix}, \tag{15}$$

so that

$$\left(J^D\right)^{-1} = \begin{pmatrix} \left(S^D\right)^{-1} & 0 \\ -\left(J^D_{cc}\right)^{-1} J^D_{cp}\left(S^D\right)^{-1} & \left(J^D_{cc}\right)^{-1} \end{pmatrix} \begin{pmatrix} I_{nb\times nb} & -J^D_{pc}\left(J^D_{cc}\right)^{-1} \\ 0 & I_{nb\times nb} \end{pmatrix}, \tag{16}$$

where $S^D = J^D_{pp} - J^D_{pc}\left(J^D_{cc}\right)^{-1} J^D_{cp}$, is the Schur complement of J^D with respect to J^D_{cc}. If $r^D = \left(r^D_n, r^D_w\right)^t$ is a given residual, the inexact action of the partitioned blocks associated to (16) can be described as follows.

Algorithm 5 (Block solving)

1. Solve $J^D_{cc} q = r^D_w$ and denote its solution by $\hat{q}$.
2. $w = r^D_n - J^D_{pc}\hat{q}$.
3. Solve $S^D p = w$ and denote its solution by $\hat{p}$.
4. $y = r^D_w - J^D_{cp}\hat{p}$.
5. Solve $J^D_{cc} s = y$ and denote its solution by $\hat{s}$.
6. Return $(\hat{p}, \hat{s})$.

If steps 1, 3 and 5 are solved iteratively instead of via a direct method, we obtain a two-stage method. Obviously, the convergence of the whole procedure depends heavily upon the convergence of each individual inner solve. Regarding this as a preconditioner, its efficiency is dictated by the way in which tolerances are chosen and satisfied for every new outer iteration.

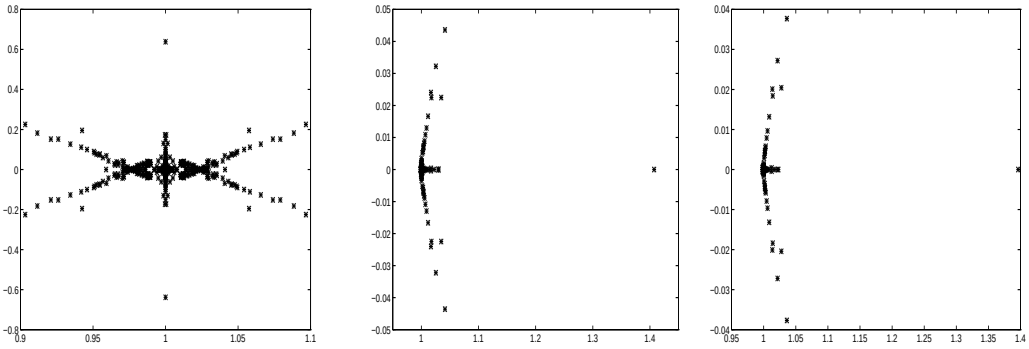

Fig. 7. Spectra of the Jacobian preconditioned on the right side by the exact version of the 2SBJ (LEFT), by the 2SGS (CENTER) and by the 2SDP preconditioners (RIGHT).

Clearly, a preconditioner like this is costly to implement in our context. It demands the solution of three linear systems with S^D probably dense. However, under this presentation it is straightforward to devise the steps for carrying out the action of different approximations to $(J^D)^{-1}$. Discarding the first step of Algorithm 5 and assuming the blocks J_{pc} and J_{cp} to be zero matrices we obtain the two-stage block Jacobi (2SBJ) preconditioner whereas the two-stage Gauss-Seidel (2SGS) results from neglecting only the block J_{pc}. This reduces Step 3 of Algorithm 5 for both 2SBJ and 2SGS to the solution of $J^D_{pp}p = r^D_n$.

These preconditioners (for exact solution of the block subsystems) yield the spectra shown in Figure 7 (left and center graphs). The left graph shows a significant clustering of the eigenvalues around the complex point $(1, 0)$ produced by 2SBJ. Not surprisingly, the 2SGS preconditioner does an even better job of clustering the eigenvalues except for one that appears separated from the rest as shown on center graph. Also, its action is similar to that of the 2SMult although the latter leaves one eigenvalue on the left half of the complex plane. This fact illustrates the close relationship between these preconditioners.

A more robust preconditioner can be obtained by means of a better approximation to the Schur complement, S, where all blocks of the original matrix are involved. In order to do this at reasonable computational costs, it is customary to provide a simple approximation to $(J^D_{pp})^{-1}$. Strategies involving the Schur complement have been employed in several linear solver variants. In CFD problems, many segregated-type algorithms work under this concept. A classical example is the Uzawa algorithm which solves the Schur complement with respect to velocity coefficients by the Richardson iteration. In contrast to flow in porous media applications, the global discretized equation is never assembled and solved in its entirety for fully implicit formulations. Many variations are possible (see e.g., [21]) ranging from solving separately for each nodal unknown to solving simultaneously for all the degrees of freedom associated with one or some (but not all) of the primary unknowns. Among the

several variants, we construct a third preconditioner inspired by the discrete projection method proposed by Turek [28] to solve saddle point formulations arising from the discretization of Navier-Stokes equations for incompressible flow. The algorithm departs from an approximation to the Schur complement with respect pressures and solves iteratively the hyperbolic component given by the velocities (role represented by concentrations in our case).

Algorithm 6 (Two-stage Discrete Projection)

1. Set $\left(\bar{J}_{cc}^{D}\right)^{-1} \simeq (J_{cc}^{D})^{-1}$.
2. Solve $\left[J_{pp}^{D} - J_{pc}^{D}\left(\bar{J}_{cc}^{D}\right)^{-1} J_{cp}^{D}\right] v_p = r_n^D - J_{pc}^{D}\left(\bar{J}_{cc}^{D}\right)^{-1} r_w^D$. Obtain $\hat{v}_p$.
3. Solve $J_{cc}^{D} v_c = r_w^D - J_{cp}^{D}\hat{v}_p$. Obtain $\hat{v}_c$.
4. Return $(\hat{v}_p, \hat{v}_c)^t$, i.e., the preconditioned residual corresponding to (r_n, r_w).

The idea behind this preconditioner is to give a sharper solution to pressures given some approximation to concentration coefficients. We propose this algorithm based on the Schur complement with respect concentrations since it is more closely related to the IMPES philosophy than the Schur complement with respect pressures which would resolve more accurately the concentration components. Throughout this work, we refer to this preconditioner as discrete projection two-stage preconditioner (2SDP).

The first step in Algorithm 6 is introduced to avoid an additional and costly iteration to solve $J_{cc}^{D} q = r_w^D$, as suggested in Step 1 of Algorithm 5. The operator $\bar{J}_{cc}^{-1}$ is chosen to be computationally cheap. Turek [28] suggests that $\bar{J}_{cc}^{-1}$ be the inverse of the diagonal part of J_{cc} (i.e., Jacobi preconditioner) which clearly reduces to the identity matrix in our case. The eigenvalue spectrum thus generated is shown in Figure 7 (right graph).

5.5 Relation between alternate and consecutive forms

There is a clear relation between the 2SAdd and the 2SBJ preconditioners and between the 2SMult and the 2SGS preconditioners. In the first step of Algorithm 3, the solution of the system is equivalent to that we would have obtained in terms of J_{pp}^{D}. In fact,

$$\begin{aligned} \left(R_p^t \widetilde{J}^D R_p\right) p = R_p^t \widetilde{D}^{-1} r \Leftrightarrow \left(R_p^t P D^{-1} J P^t R_p\right) p = R_p^t P D^{-1} P^t r \\ \Leftrightarrow J_{pp}^{D} p = r_n^D \end{aligned} \tag{17}$$

Similarly, we can obtain the same correspondence for the concentrations. Once a solution for both types of variables is computed, the alternate two-stage

preconditioners proceed to improve the residuals by a correction with the preconditioner $\hat{M}$.

It follows that the error propagation operator associated to the 2SComb preconditioner is given by

$$\begin{aligned} E_{\text{comb}} &= I - \hat{J} M_p^{-1} \\ &= \left(I - \tilde{J}\hat{M}^{-1}\right)\left(I - \tilde{J} R_p \left(R_p^t \tilde{J}^{W_p} R_p\right)^{-1} R_p^t \widetilde{W}_p\right). \end{aligned}$$

In a similar way, we can get expressions for the additive and multiplicative extensions,

$$E_{\text{add}} = \left(I - \tilde{J}\hat{M}^{-1}\right)\left[I - \tilde{J}\left(t_p + t_c\right)\right],$$

$$E_{\text{mult}} = \left(I - \tilde{J}\hat{M}^{-1}\right)\left[I - \tilde{J}\left(t_p + t_c - t_c \tilde{J} t_p\right)\right].$$

By (17) we obtain the 2SBJ and 2SGS error propagation operators as

$$E_{\text{BJ}} = I - J M_{\text{BJ}}^{-1} = I - \tilde{J}^D\left(t_p + t_c\right),$$

$$E_{\text{GS}} = I - J M_{\text{GS}}^{-1} = I - \tilde{J}\left(t_p + t_c - t_c \tilde{J}^D t_p\right).$$

Note that even if we drop the correction step (given by $\hat{M}$) from the alternate two-stage preconditioners, there is no way to reproduce the same relative residual convergence history within the iterative solver since the alternate type acts on the coupled system whereas the consecutive type acts on the already decoupled system which is expected to be easier to solve.

As shown above, the error propagation associated to $\hat{M}$ is a factor in the error propagation associated to the whole two-stage preconditioner. To ensure convergence, the norm of each error should be bounded above by 1. This imposes the same restrictions on each of the factors involved in the complete error propagation operator. Moreover, a high error propagation norm (one marginally close to 1) should be compensated by a low error propagation from the other factor in order to get faster convergence rates. It is at this point that we find a serious limitation, not to say a drawback, in the use of $\hat{M}$ to correct residuals. This situation seems to be more severe as the problem size or inherent complexity of the problem increases. To put things in perspective, we can mention a couple of facts:

– There is an inherent penalty in introducing the operator $\hat{M}$. The computation of new residuals involves one extra matrix vector multiplication and an AXPY operation. This can certainly be computationally demanding for

large scale problems and for iterative solvers that perform more than one call to the preconditioner per iteration (e.g., BiCGSTAB, CGS).

- Decoupling operators are effective in clustering the eigenvalues of the original highly indefinite Jacobian. The combinative, additive and multiplicative reimpose this task on the operator $\hat{M}$. For instance, we require

$$\|E_{\text{add}}\| \leq \left\|I - \tilde{J}\,(t_p + t_c)\right\| \left\|I - \tilde{J}\hat{M}^{-1}\right\| \leq 1,$$

for the two-stage additive error propagation operator. If $t_p + t_c$ does a good job of preconditiong $\tilde{J}$, $\hat{M}$ should be formulated to have a better or at least, a comparable effect. Note that the omission the decoupling operation leaves one with the difficult task of finding an efficient global preconditioner $\hat{M}$ for the original Jacobian matrix, which is additionally expected to eliminate those low error frequencies remaining from pressure and concentration iterations. Not surprisingly, the spectrum plots for the alternate two-stage preconditioner are less compact than their consecutive counterparts.

Of course, a more elaborated $\hat{M}$ may eventually provide the desired effectiveness but at a significantly higher cost. Although the use of the operator $\hat{M}$ seems to be better justified in the original combinative method, it still has to capture part of the hyperbolic behavior contained in concentrations from a linear problem whose block algebraic properties make it difficult (recall spectrum pictures in Figures 3-6), instead of taking advantage of an easier reduced concentration problem obtained by a better decoupling strategy.

The last point can be made more precise by looking at the computation of

$$\begin{aligned} E_{\text{add}} &= \left(I - \tilde{J}\hat{M}^{-1}\right)\left[I - \tilde{J}\,(t_p + t_c)\right] \\ &= \left(I - \tilde{J}\hat{M}^{-1}\right)\left[I - \tilde{J}^D\,(t_p + t_c) - \left(\tilde{J} - \tilde{J}^D\right)(t_p + t_c)\right]. \end{aligned}$$

Consequently, by taking norms we obtain

$$\|E_{\text{add}}\| \leq \gamma\,(\|E_{\text{BJ}}\| + \eta)\,, \tag{18}$$

where $\gamma = \left\|I - \tilde{J}\hat{M}^{-1}\right\|$ and $\eta = \left\|\left(\tilde{J} - \tilde{J}^D\right)(t_p + t_c)\right\|$. Equation (18) shows the penalty introduced by η into the final error estimate as a result of leaving out the preconditioning effect of the decoupling operator. Additionally, this variable is unlikely to be smaller than $\|E_{\text{BJ}}\|$. The variable γ has to compensate this penalty by preconditioning effectively with $\hat{M}$ the original coupled linear system and decreasing the overall error propagation factor.

The use of $\hat{M}$ seems to be only justified in special cases. For example, it can be used for retrieving part of the global information lost in a line correction method. Other acceptable form could be a coarse representation of the original discretization. However, reliable coarse meshes for hyperbolic problems are

not easy to obtain creating a problem for enhancing concentration residuals. In general, it should be designed under simple terms on sequential implementations and with more relaxed bounds if it is intended for vector and parallel implementations. We believe that better results at lower computational demands can be obtained by incorporating more information contained in the decoupled blocks and improving the performance of each subsystem solution.

Finally, we remark that Fan *et al.* just recently arrived at similar observations in the context of microelectronic device simulation [19]. They experimentally observed that further preconditioning after the decoupling stage with the ABF transformation proposed by Bank *et al.* resulted in significant improvement compared to block preconditioning alone.

6 Computational experiments

We include both sequential and parallel implementations. In the first one, we compare the six preconditioners discussed in this work and three other "classical" ones which do not use any decoupling strategy. In the parallel implementation part, we only compare the 2SComb (one of the best preconditioners known so far in the reservoir community) and 2SGS preconditioners (which incorporates full decoupling strategy as a preconditioning stage).

6.1 Description of test case

Table 1
Physical input data.

Initial nonwetting phase pressure at 49 ft	$300psi$
Initial wetting saturation at 49 ft	.5
Nonwetting phase density	$48lb/ft^3$
Nonwetting phase compressibility	$4.2 \times 10^{-5}psi^{-1}$
Wetting phase compressibility	$3.3 \times 10^{-6}psi^{-1}$
Nonwetting phase viscosity	$1.6cp$
Wetting phase viscosity	$0.23cp$
Areal permeability	$150md$
Permeability along 1st and 2nd half of vertical gridblocks	$10md$ and $30md$

Table 1 summarizes the physical parameters for this problem, and Figure 8 shows the relative permeability and capillary pressure functions used. The

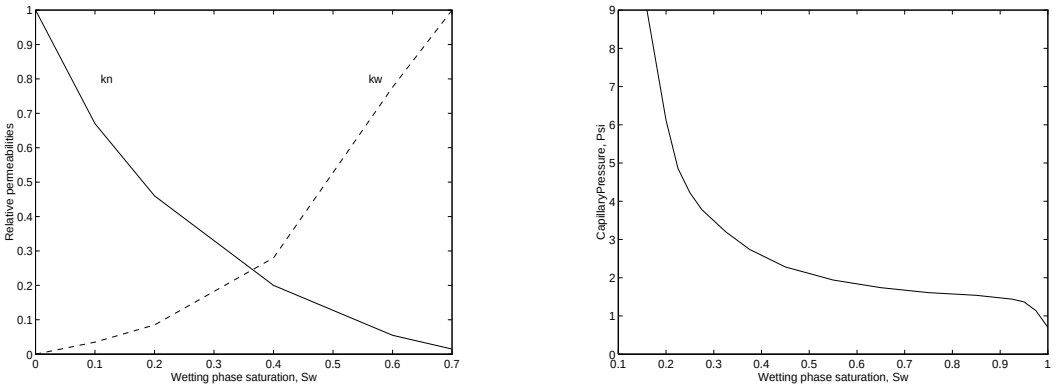

Fig. 8. Relative permeability of both phases (LEFT) and Capillary pressure function (RIGHT).

model consists of a water injection well located at the coordinate $(1, 1)$ of the plane and a production well (both vertical and with bottomhole pressure specified) at the opposite corner of the plane. The permeability is uniform in the areal sense and 5-15 times higher than that in the vertical direction.

6.2 Sequential implementation case

We use non-uniform grid spacing and two different discretization sizes: $8 \times 8 \times 4$ and $16 \times 16 \times 4$. We ran both cases with time steps $\Delta t = 0.1, 1.0$ days. The data for the tests was downloaded from the simulation after 1 time step and after 3 Newton iterations within the current time level. The code including all the combinations of linear solver and preconditioner tested was written in FORTRAN 77 and all of the tests were run on a single node of an IBM SP1(RS6000, model 370, with a 62.5 MHz clock). These nodes give a peak performance of 125 MFlops and have 128 MB of RAM.

The tests included runs made with both GMRES and BiCGSTAB preconditioned with each of the schemes analyzed and, additionally, with three preconditioners of common use in reservoir simulation (particularly the last two), i.e., tridiagonal, ILU(0) (incomplete LU factorization with no infill) and block Jacobi. Table 2 shows the results for all the preconditioners applied to GMRES and Table 3 shows the corresponding results for BiCGSTAB preconditioned with each of the schemes. Each of these tables has results for the four possible combinations of time step size and spatial discretization size. The four columns of each of these sections on both tables list the number of outer linear iterations, N_{it}, the total elapsed time for the iteration of the solver, T_s, the elapsed time incurred in setting up the preconditioner, T_p, and the average number of inner linear iterations per one outer iteration, $N_{i,a}$, respectively from left to right.

Table 2
GMRES Results. Preconditioners shown from top to bottom: Tridiagonal, Incomplete LU factorization with no infill, Block Jacobi, 2SComb, 2SAdd, 2SMult, 2SBJ, 2SGS, 2SDP.

Time Step Size →		$\Delta t = .1$				$\Delta t = 1.$			
	Precond.	N_{it}	T_s	T_p	$N_{i,a}$	N_{it}	T_s	T_p	$N_{i,a}$
	Tridiag.	782	12.16	0.26		>1000	-	-	
Prob.	ILU(0)	859	109.01	0.37		>1000	-	-	
Size	BJ	363	7.60	0.17		358	7.57	0.18	
4	2SComb.	390	227.50	0.63	42	>1000	-	-	
×	2SAdd.	42	83.38	1.00	121	30	79.00	0.88	200
8	2SMult.	19	38.38	1.00	120	18	47.63	0.88	195
×	2SBJ	32	19.54	0.02	123	21	24.41	0.02	198
8	2SGS	16	9.53	0.02	119	13	15.42	0.02	192
	2SDP	15	15.80	0.03	122	11	20.39	0.04	188

Time Step Size →		$\Delta t = .1$				$\Delta t = 1.$			
	Precond.	N_{it}	T_s	T_p	$N_{i,a}$	N_{it}	T_s	T_p	$N_{i,a}$
	Tridiag.	>1000	-	-		>1000	-	-	
Prob.	ILU(0)	840	141.22	5.51		>1000	-	-	
Size	BJ	170	152.51	37.18		424	381.22	37.44	
4	2SComb.	555	527.25	9.63	15	>1000	-	-	
×	2SAdd.	237	680.00	13.50	52	384	1678.13	13.50	80
16	2SMult.	103	305.63	13.50	52	90	392.75	13.50	79
×	2SBJ	52	58.66	0.09	52	37	61.77	0.09	80
16	2SGS	25	28.77	0.09	52	19	31.68	0.09	78
	2SDP	21	49.91	0.13	63	17	52.27	0.13	85

All of the linear systems were solved iteratively until a norm reduction of 1×10^{-6} was achieved, relative to the initial one given by the 2-norm of the right hand side since zero was used as the initial guess in every case. GMRES(30), with tridiagonal preconditioning and a linear tolerance equal to that of the outer solves, was used as the two-stage inner solver.

Some general comments of the results are in order. The traditional preconditioners do not take into account any of the physics of the multi-phase model

Table 3
BiCGSTAB Results. Preconditioners shown from top to bottom: Tridiagonal, Incomplete LU factorization with no infill, Block Jacobi, 2SComb, 2SAdd, 2SMult, 2SBJ, 2SGS, 2SDP.

Time Step Size →		$\Delta t = .1$				$\Delta t = 1.$			
	Precond.	N_{it}	T_s	T_p	$N_{i,a}$	N_{it}	T_s	T_p	$N_{i,a}$
	Tridiag.	127	3.42	0.26		227	6.20	0.27	
Prob.	ILU(0)	239	57.90	0.37		>1000	-	-	
Size	BJ	80	2.98	0.17		75	2.83	0.17	
4	2SComb.	106	113.75	0.50	40	125	253.50	0.50	81
×	2SAdd.	24	88.75	1.00	118	34	158.38	0.75	183
8	2SMult.	13	61.38	1.00	115	14	67.38	0.75	188
×	2SBJ	23	24.97	0.02	116	24	46.91	0.02	173
8	2SGS	11	11.91	0.02	115	12	24.15	0.02	177
	2SDP	10	20.04	0.03	118	14	44.01	0.03	179

Time Step Size →		$\Delta t = .1$				$\Delta t = 1.$			
	Precond.	N_{it}	T_s	T_p	$N_{i,a}$	N_{it}	T_s	T_p	$N_{i,a}$
	Tridiag.	176	43.37	5.54		>1000	-	-	
Prob.	ILU(0)	>1000	-	-		424	381.22	37.44	
Size	BJ	57	118.53	45.99		69	115.64	37.46	
16	2SComb.	170	292.50	9.75	14	180	690.00	12.00	25
×	2SAdd.	68	361.75	13.38	50	61	490.63	13.18	77
16	2SMult.	44	238.88	13.38	49	32	255.25	13.38	75
×	2SBJ	41	83.21	0.09	49	23	68.91	0.09	76
4	2SGS	17	35.82	0.09	50	11	33.81	0.09	76
	2SDP	13	56.72	0.12	61	10	58.62	0.12	82

and either fail to converge or are outperformed by some of the more thoughtful preconditioners, as the trend suggests when the spatial discretization is refined. Notice that neither of the two problem sizes tested here are anywhere near the size of numerical models that the reservoir simulation community wishes to tackle in today's high performance computing environment. In particular, ILU(0) fails to resolve the low error frequencies. Moreover, the block Jacobi preconditioner appears to be the more reliable one of the traditional kind. However, our implementation of block Jacobi inverts directly four blocks

of the Jacobian matrix. Such a rich block Jacobi preconditioner may not be realizable in practice, specially in parallel implementations where each of the blocks should live in one processor to minimize the communication overhead.

Turning to a more detailed analysis of the results, the timings of BiCGSTAB and GMRES are comparable in spite of the lower number of outer iterations given by BiCGSTAB. This owes to the fact that BiCGSTAB has two matrix-vector multiplies per iteration instead of the single one needed by GMRES. Additionally, the convergence of BiCGSTAB is erratic, as is well known and can be appreciated in Figure 10.

Comparison between the results for $\Delta t = 0.1$ and those for $\Delta t = 1.0$ shows a greater number of outer iterations for the first four preconditioners (with a few exceptions) for the longer time step. However, all of the two-stage (except for the 2SComb) preconditioners give a smaller number of outer iterations for the longer time step. The key in interpreting these results is in the action of the full-decoupling operator implemented for the two-stage preconditioners and its own power to precondition the system. We believe that the *weight* of the off-diagonal Jacobian blocks after full decoupling is less for the longer time step than for the shorter one and the preconditioner is more effective as a result. To this point, notice that the combinative preconditioner, which only uses partial decoupling shows a greater number of outer iterations for $\Delta t = 1.0$ than that for the shorter Δt. The increased difficulty of the problem with a longer time step is reflected in all cases by the growth in the average number of inner iterations per unit outer iteration.

The number of inner iterations per step of the outer iteration is comparable in the results for both iterative solvers, except for minor differences due to particular convergence history of each case. Notice that in the case of the last five preconditioners $N_{i,a}$ shows the accumulated average of both the pressure and concentration components whereas 2SComb only solves for pressure components and therefore show a lower number of inner iterations. An increase in the time step size damages the diagonal dominance of the main-diagonal blocks of the decoupled Jacobian thus producing harder inner solves, as reflected by the results on both tables. Somehow surprisingly, a growth in the size of the linear system decreased $N_{i,a}$ in every case.

As for the question of efficiency, the consecutive-type preconditioners, i.e., the two-stage block Jacobi, Gauss-Seidel and discrete projection, display the best elapsed times to converge the linear systems typical in fully implicit black-oil simulation. As mentioned above, although the problem sizes presented here are only modest, the consecutive preconditioners appear to have the required robustness for problems of greater size. The combinative preconditioner, for example, is not robust enough even for these rather friendly problems.

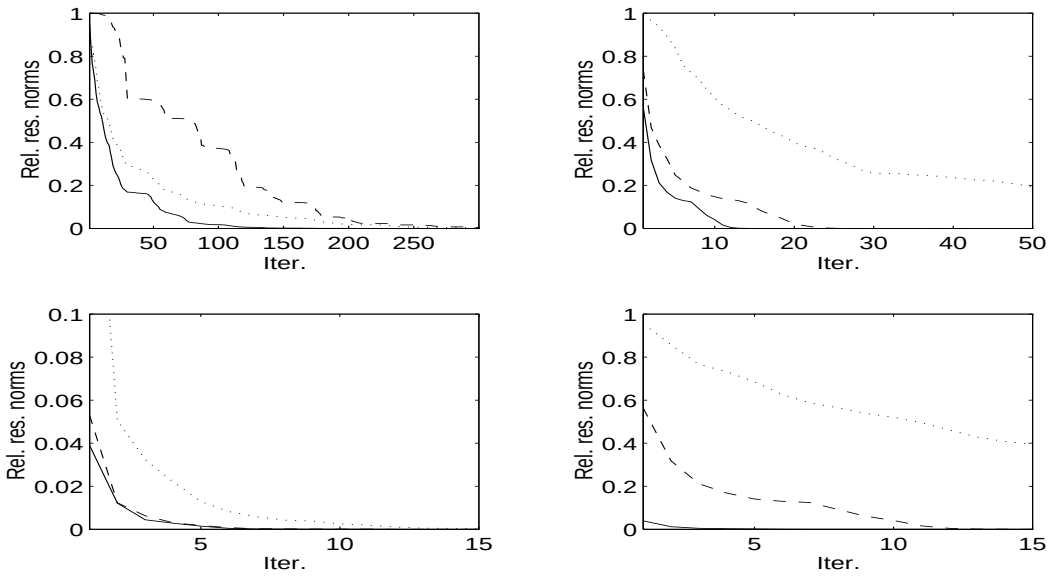

Fig. 9. Relative residual norms vs. iterations of preconditioned GMRES. Subplot $(1,1)$: ILU(0) (dot), Tridiag(dash), block Jacobi (solid). Subplot $(1,2)$: 2SComb (dot), 2SAdd (dash), 2SMult (solid). Subplot $(2,1)$: 2SBJ (dot), 2SGS (dash), 2SDP (solid). Subplot $(2,2)$: Block Jacobi (dot), 2SMult (dash), 2SDP (solid). Problem Size: $4 \times 8 \times 8$. $\Delta t = 0.1$.

According to the trend suggested by outer iterations, the 2SDP appears as the best preconditioner (although, 2SGS achieves similar results). However, this appreciation can be misleading when looking at overall timings. Note that 2SDJ performs more inner iterations on average than its closer competitor 2SGS. The reason is that the Schur complement matrix with respect concentrations is more poorly conditioned than the pressure and concentration blocks resulting from the decoupling process. Although, pressure and concentration blocks are M-matrices (recall Theorem 1) there is no guarantee that Schur complement systems or approximation to them are well behaved in this case. In all experiments, we use the identity matrix to approximate $\bar{J}_{cc}^{D}$ in the construction of the Schur complement with respect to concentrations.

A final word is devoted to the comparison of the alternate with the consecutive preconditioners. The former family is approximately equivalent to the latter but with the addition of the global preconditioning step given by $\hat{M}$ (this step is absent in the consecutive type). The total elapsed times testify to the high overhead incurred in the application of the global preconditioner of the alternate schemes. Moreover, as was mentioned above, $\hat{M}$ should be at least as effective as a preconditioner as are the individual decoupled pressure and concentration blocks. However, $\hat{M}$ is a preconditioner for the full Jacobian, which throws us back to beginning of the path, or worse. We are now looking for a preconditioner for J that has to beat the action of the decoupled blocks. These experiments show clearly that this is a loosing proposition and therefore, the application of $\hat{M}$ results in wasted time by the iterative solver. It should

be mentioned that $\hat{M}$ was chosen as an LU factorization of J with complete infill inside a bandwidth of 19. The number of bands was chosen so that the coupling of nearest-neighbor layers was always retained (notice that all cases have 4 grid blocks in the z-direction, which is most rapidly increasing in the numbering scheme of the grid blocks). In spite of the assumed robustness of this global preconditioner, its main effect seems to be the posting of greater elapsed times T_s with no reduction in the number of linear iterations.

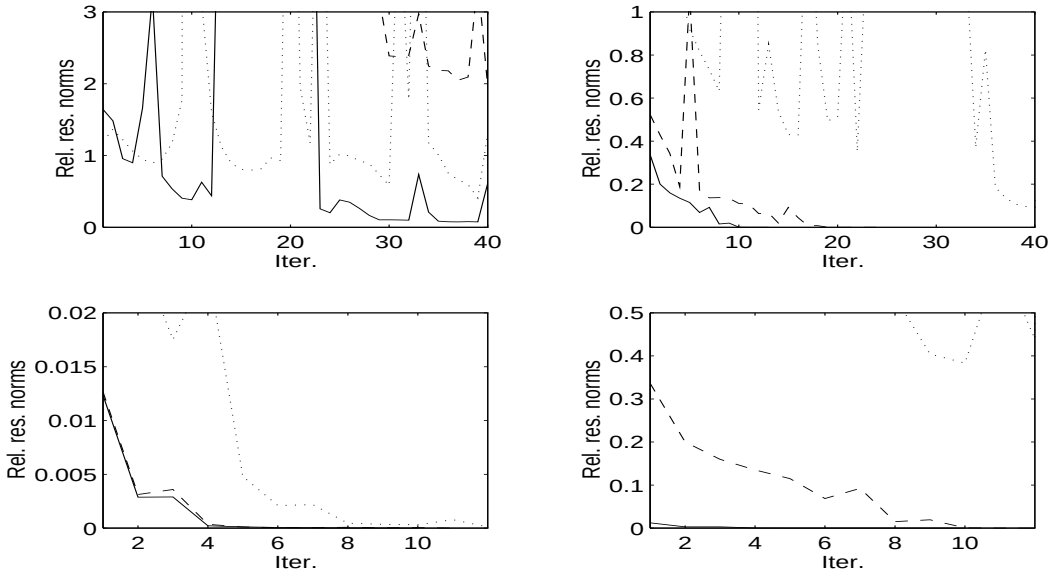

Fig. 10. Relative residual norms vs. iterations of preconditioned BiCGSTAB. Subplot $(1,1)$: ILU(0) (dot), Tridiag(dash), block Jacobi (solid). Subplot $(1,2)$: 2SComb (dot), 2SAdd (dash), 2SMult (solid). Subplot $(2,1)$: 2SBJ (dot), 2SGS (dash), 2SDP (solid). Subplot $(2,2)$: Block Jacobi (dot), 2SMult (dash), 2SDP (solid). Problem Size: $4 \times 8 \times 8$. $\Delta t = 0.1$.

We mentioned in the algebraic analysis of the Jacobian matrix blocks that the properties we require from the individual blocks are met for reasonable time step sizes, i.e., clearly our assumptions will not be all valid for Δt beyond a given threshold value. We also remarked that the two-stage combinative preconditioner was expected to deteriorate as the time step size increases because the pressure components (which it is based on) are no longer dominant. The results of these experiments show that this is the case while not noticeably affecting any of the other two-stage schemes. In view of these results we believe there is still considerable room in choosing a Δt which will guarantee convergence of the Newton method itself without substantially damaging the performance of any of the five newly proposed two-stage schemes.

Figure 9 summarizes the convergence behavior of GMRES for the discretization size of $8 \times 8 \times 4$ and $\Delta t = 0.1$. On the upper left corner, the plot shows the results for the three standard preconditioners. The plot on the upper right shows the convergence of the alternate preconditioners and the one on the

lower left corner shows the results for the consecutive schemes. The remaining plot on the lower right corner shows the best results out of each of the other three plots. Figure 10 shows the same exact arrangement for BiCGSTAB. We note that as the effectiveness of the preconditioner increases, the characteristic erratic behavior of BiCGSTAB gets damped. The results for GMRES testify to its robustness and efficiency when preconditioned by two-stage methods, specially of consecutive type. Note, on figure 9, lower-right plot, that the 2SGS consecutive preconditioner produces a dramatically faster convergence than does the fastest of the alternate preconditioners.

6.3 Parallel implementation case

We test parallel implementations of the 2SComb (used with relative success in reservoir simulation [31]) and 2SGS preconditioners, attached to the Newton scheme of the simulator described above. The parallel 2SComb preconditioner includes a z-line correction (see e.g., [22]) and GMRES solution of the 2D inner system, with block Jacobi preconditioner, giving it some advantage not exploited in the sequential implementation. The line correction method in the 2SGS has some difficulties due to the lack of diagonal dominance of the pressure block matrix when there are relative small capillary pressure gradients compared to permeability gradients of the wetting phase at large time steps, thus violating the conditions of Theorem 1. (This situation does not arise in the concentration coefficient block but this 3D system is really easy to solve iteratively anyway.) Since the line-search backtracking method helps handle guesses far from the nonlinear solution, we reinforce the preconditioner robustness by using GMRES to solve both 3D inner problems, with block tridiagonal preconditioner, in order to be able to take larger time steps. Overall, we estimate one linear iteration using the 2SGS scheme to be about 40% more expensive than using the 2SComb.

Numerical experiments are performed on an Intel Paragon and on an IBM SP2 parallel machines. The Paragon machine has 42 nodes arranged in a 2-D mesh topology, 64 Mbytes of RAM plus 16 Kbytes of data cache and a peak performance of 80 Mflops per node, with a peak interprocessor transfer rate of 40 Mbytes/s. The SP2 machine consists of 16 nodes, each with 128 Mbytes of RAM and peak performance of 260 MFlops and a bidirectional communication rate of 50 MBytes/s. We use the MPICH message passing system library for portability of the simulator.

The data are decomposed in an areal sense (i.e., each processors holds the same original number of gridblocks along the depth direction), since in most reservoir domains the vertical dimension is relatively much smaller than the horizontal plane. The effective manipulation of a full permeability tensor in-

Table 4
Summary of linear iterations (LI), nonlinear iterations (NI), number of backtracks (NB) and execution times of GMRES and Bi-CGSTAB with 2SComb and 2SGS. Simulation of 20 time steps with $\Delta t = .05$ and $\Delta t = .5$ Problem size: $8 \times 24 \times 24$ gridblocks. Intel Paragon processor mesh of 4×4 nodes. (*): Backtracking method failed after the $17th$ time step; (**): Δt was halved after the $16th$ time step.

Δt	Linear solver/Prec.	LI	NI	NB	Time(Hrs.)
.05	GMRES/2SComb	1450	45	0	1.10
	GMRES/2SGS	102	49	0	0.11
	Bi-CGSTAB/2SComb	855	45	0	1.19
	Bi-CGSTAB/2SGS	66	44	0	0.07
.5	GMRES/2SComb	6745	100	0	6.37
	GMRES/2SGS	538	107	0	0.51
	Bi-CGSTAB/2SComb(*)	2808	190	41	5.62
	Bi-CGSTAB/2SGS (**)	493	102	12	0.70

duces a 19-point stencil discretization for the pressures and concentrations of the linearized wetting phase equation and, a 19-point stencil for pressures and a 7-point stencil for concentrations of the linearized non-wetting phase equation (this gives rise to the 64 coefficient arrays accompanying each gridblock unknown). Therefore, matrix-vector products involve data communication of each node with its four lateral and four corner neighbors [12].

Table 4 (Intel Paragon) shows that both GMRES and BiCGSTAB algorithms perform similarly for a problem of modest difficulty (i.e., for $\Delta t = .05$). Notice that BiCGSTAB employs almost half of the total number of iterations of GMRES but, on the other hand, BiCGSTAB doubles the number of matrix-vector multiplications and preconditioner calls made by GMRES at each linear iteration, making the times comparable between these two linear solvers. In simple problems, Bi-CGSTAB can outperform GMRES, whereas in more complex problems the latter tends to be more robust and efficient as the simulation for $\Delta t = .5$ reveals.

Also remarkable is the performance of both linear solvers with the 2SGS preconditioner in relation to the 2SComb preconditioner. For this particular problem, the 2SGS preconditioner reduces by more than a 10-fold the total number of linear iterations. Since the number of nonlinear iterations is practically unchanged, we improve the computer times by almost 10 times (recall discussion on cost of both schemes). This result corroborates the observations made in the previous section for sample matrices extracted from this physical model.

BiCGSTAB fails twice for different reasons. For $\Delta t = .5$, the 2SGS precondi-

tioner forces a reduction of the time step due to the high changes of pressures and concentrations within the time step. In many reservoir simulation codes it is customary to regulate the next time step according to a maximum allowable change of pressures and concentrations within the current time step. This prevents possible loss of material balance due to the deterioration or eventual failure of the nonlinear solution. Shortening the time step increases the chances of convergence for the nonlinear method. The failure with the 2SComb preconditioner is more serious. The line-search failed because the linear solver was unable to converge at the maximum tolerance allowed (0.1, in our case). Therefore, BiCGSTAB could not provide an acceptable direction for decreasing $\|F\|$. Note that, before breakdown, this execution had undergone a high number of backtracks and nonlinear steps.

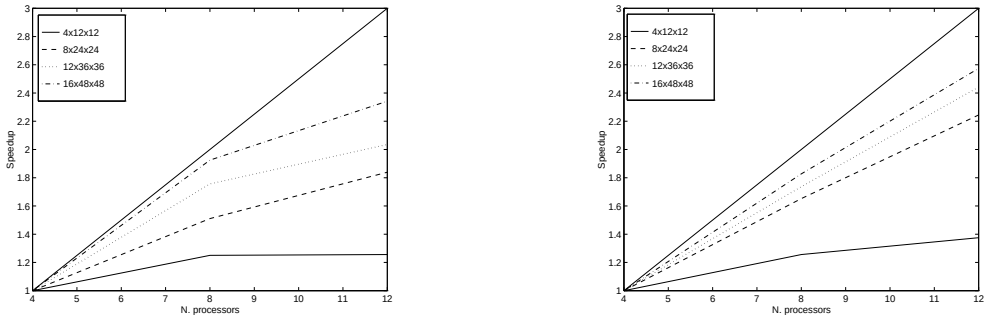

Fig. 11. Speedup vs. number of processors for the two-phase problem using the Newton/2SGS solver on an Intel Paragon (LEFT) and an IBM SP2 (RIGHT) after 20 time steps.

Figures 11 and 12 summarize parallel scaling on the Paragon and IBM SP2. Note that speedups are normalized to 4-processor runs. For four different problem sizes, we compare timings on 4, 8 and 12 processors. This was enough to capture the efficiency trend of the Newton/2SGS solver on both machines. The simulator scales better on the IBM SP2 than on the Intel Paragon, mainly due to the lower latency and higher bandwidth of the former. As expected, the larger the problem size the greater the efficiencies obtained. Note how the computing time for the smallest problem size is practically governed by the communication overhead in both machines.

The major bulk of parallelism resides in the computation of the preconditioner. The block tridiagonal preconditioner used in the innermost GMRES is totally parallel and sufficient to meet the required linear tolerances. Unfortunately, most of the operations in the inner and outer GMRES are parallelizable at the level of BLAS-1 operations (i.e., AXPY's and inner products). In this regard, the classical Gram-Schmidt is chosen over the modified Gram-Schmidt to exploit further parallelism in the construction of the Krylov basis without sacrificing stability requirements. (However, the GMRES implementation contemplates iterative refinement to preserve orthogonality if required.)

Having in mind the above discussion, the scaling results shown in both figures are somehow encouraging in spite of the significant portion of code with a small degree of fine grain parallelism. The results on the Paragon show a more rapid speedup degradation but at the same time display a greater efficiency gain as the problem size increases than the SP2. This is explained by the fact that interprocessor communication on the Paragon is latency-bound because of the relatively long times needed to set up a message of zero length. The much shorter latencies of the SP2 make the interprocessor transfer time more linearly dependent on message length, thus keeping the ratio of computation to communication fairly constant for different problem sizes.

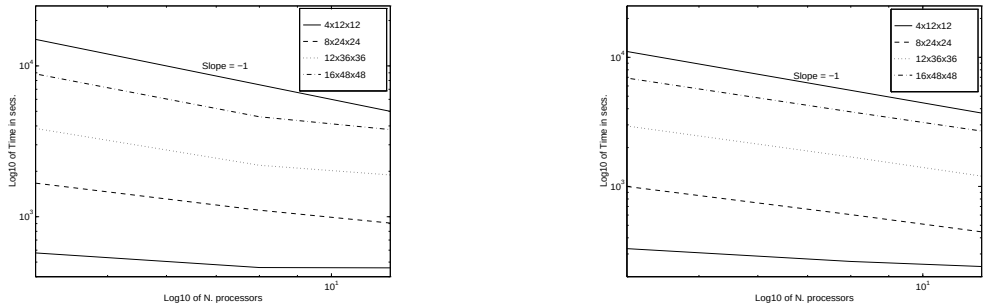

Fig. 12. Log-log plot of the number of processors vs. execution time for the two-phase problem using the 2SGS preconditioner on an Intel Paragon (LEFT) and on an IBM SP2 (RIGHT) after 20 time steps.

The log-log plots in Figure 12 show the deviation from ideal speedup (indicated with slope -1) that all problem size cases present on both the Paragon and the SP2. This complements the observation that timings are less sensitive to degradation as more processors are added in the SP2. In these experiments the SP2 shows to be from 50% to 100% faster than the Paragon. In theory this margin is expected to be larger, but the author suspects that memory hierarchy effects may be deteriorating the performance of the SP2.

Figure 13 (left graph) illustrates the relatively strong impact of the 2SGS preconditioner on the simulation. For a moderate problem size, GMRES with 2SComb preconditioning takes over 10 times more linear iterations than with 2SGS preconditioning. The difference between the 2SComb and 2SGS preconditioning is less prominent in terms of computer time (right graph). The line correction in the 2SComb preconditioner contributes to reducing the cost for solving the pressure system since, at the beginning of this simulation, the capillary pressure gradients are high. This method was not introduced in the 2SGS in order to preserve the highest possible robustness. Despite this the Newton/2SGS solver still outperforms by almost a three-fold the timings of the inexact Newton/2SComb solver.

The previous analysis for a particular time step explains clearly the saving of GMRES iterations for a moderately long simulation. Figure 14 (left graph)

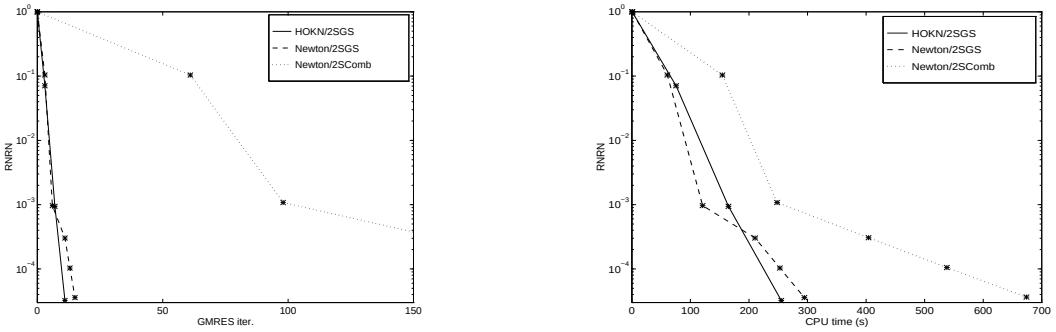

Fig. 13. Number of accumulated GMRES iterations vs. relative nonlinear residual norms (LEFT) and CPU time vs. relative nonlinear residual norms (RIGHT) using the 2SGS and 2SComb preconditioners on 12 nodes of the IBM SP2 for a problem size of $16 \times 48 \times 48$ at the third time step.

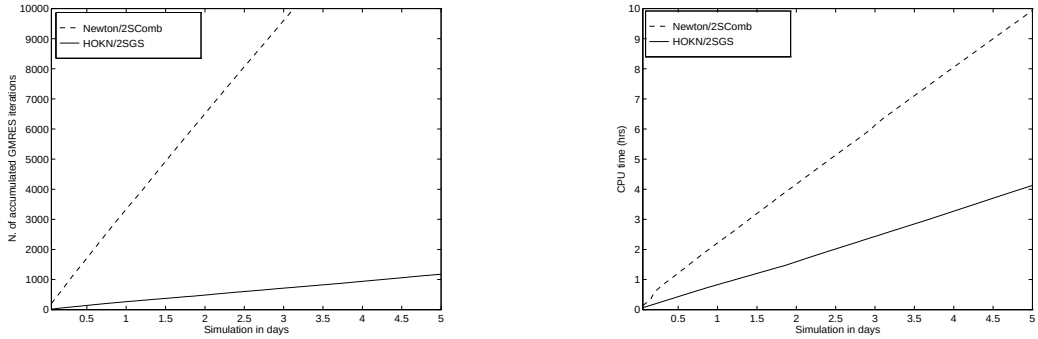

Fig. 14. Performance in accumulated GMRES iterations (LEFT) and accumulated CPU time (RIGHT) of Newton/2SGS and Newton/2SComb solvers after 100 time steps of simulation with $\Delta t = .05$ of a $16 \times 48 \times 48$ problem size on 16 IBM SP2 nodes.

shows that the new Newton/2SGS spends a considerably smaller amount of GMRES iterations compared to the Newton/2SComb solver. As before, since one GMRES iteration is more expensive with the 2SGS preconditioner than with the 2SComb preconditioner the Figure 14 (right graph) shows that simulation times are reduced by more than a three-fold with the new solver.

7 Conclusions

We have described a family of two-stage preconditioners for linear systems arising in coupled systems of nonlinear partial differential equations modeling multi-phase flow. Our analysis has shown that a simple decoupling strategy combined with inexact solution of the main blocks governing the physical process provides a powerful device to speedup the convergence of Krylov iterative methods such as GMRES and BiCGSTAB. Our results outperform a few traditional approaches proposed in the literature of reservoir engineering:

ILU(0), block Jacobi or banded preconditioner (for preconditioning of the entire system), and an inexact version of the combinative approach (originally developed for coupled linear systems).

These two-stage preconditioners have been developed under a simple and general basis: they are easy to implement and can afford the use of several efficient iterations to solve the resulting inner linear systems. We believe that this simplicity and generality provides a satisfactory vehicle to extend iterative linear solver theory already developed for individual PDE's to coupled systems of PDE's. Although the ideas presented here could be fitted into several scenarios, special considerations on the physics behind the problem may lead to further enhancements and interpretations. In our particular case, the consideration of full decoupling translated in a way to concentrate parabolic convection-diffusion information in the main diagonal blocks producing the algebraic properties that make them amenable to efficient inner iterative solvers.

Also included is a parallel implementation of the best two-stage preconditioner proposed in this work to expose its high performance capabilities. However, the possibilities here are enormous since a higher degree of scalability can be achieved by focusing on well known parallel strategies for preconditioning single phase problems (e.g., overlapping and non-overlapping Schwarz domain decomposition methods, multigrid, line-correction).

Full understanding and conception of efficient preconditioners for coupled linear systems is a difficult task. Only a few experiences have been reported on the topic even though coupled systems of equations arise in many application areas. Further theoretical investigation and computational experiments remain to be done with more PDE's coming into play, such as in the case of compositional and thermal reservoir simulation. We certainly encourage research in this direction.

Acknowledgement

The authors wish to thank Argonne National Laboratories for the use of the IBM SP1.

References

[1] J.M.C. Aarden and K.E. Karlsson. Preconditioned cg-type methods for solving coupled system of fundamental semiconductor equations. *BIT*, 29:916–937, 1989.

[2] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1994.

[3] K. Aziz and A. Sethari. *Petroleum Reservoir Simulation.* Applied Science Publisher, 1983.

[4] R.E. Bank, T.F. Chan, W.M. Coughran, and K. Smith. The alternate–block–factorization procedure for systems of partial differential equations. *BIT*, 29:938–954, 1989.

[5] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Dunato, J. Dongarra, V. Eikkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods.* SIAM, Philadelphia, 1994.

[6] G.A. Behie and P.A. Forsyth. Incomplete factorization methods for fully implicit simulation of enhanced oil recovery. *SIAM J. Sci. Statist. Comput.*, 5:543–561, 1984.

[7] G.A. Behie and P.K.W. Vinsome. Block iterative methods for fully implicit reservoir simulation. *Soc. of Pet. Eng. J.*, pages 658–668, Oct. 1982.

[8] R. Bhogeswara and J. E. Killough. Domain decomposition and multigrid solvers for flow simulation in porous media on distributed memory parallel processors. *Journal of Scientific Computing*, 7:127–162, 1992.

[9] P.E. Bjørstad, W.M. Coughran Jr., and E. Grosse. Parallel domain decomposition applied to coupled transport equations. In D.E. Keyes and J. Xu, editors, *Seventh International Conference on Domain Decomposition Methods for Scientific Computing*, Como, Italy, 1993. American Mathematical Society.

[10] P.N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Statist. Comput.*, 11:450–481, 1990.

[11] R. Bru, V. Migallon, J. Penades, and D.B. Szyld. Parallel, synchronous and asynchronous two-stage multisplitting methods. *ETNA, Electronic Transactions on Numerical Analysis*, 3:24–38, 1995.

[12] C. Dawson, H.M. Klíe, C. San Soucie, and M.F. Wheeler. A parallel, implicit, cell-centered method for two-phase flow. In preparation, 1996.

[13] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[14] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice–Hall, Englewood Cliffs, New Jersey, 1983.

[15] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact Newton method. Technical Report TR94–25, Dept. of Computational and Applied Mathematics, Rice University, 1994.

[16] S.C. Eisenstat and H.F. Walker. Globally convergent inexact Newton methods. *SIAM J. Optimization*, 4:393–422, 1994.

[17] H.C. Elman and G. H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, 31:1645–1661, 1994.

[18] L. Elsner and V. Mehrmann. Convergence of block iterative methods for linear systems arising in the numerical solution of Euler equations. *Numer. Math.*, 59:541–559, 1991.

[19] Q. Fan, P.A. Forsyth, J.R.F. McMacken, and W.P. Tang. Performance issues for iterative solvers in device simulation. *SIAM J. Sci. Statist. Comput.*, 17:100–117, 1996.

[20] G. Golub and M.L. Overton. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. *Numer. Math.*, 53:571–593, 1988.

[21] V. Haroutunian, M.S. Engelman, and I. Hasbani. Segregated finite element algorithms for the numerical solution of large–scale incompressible flow problems. *I. J. for Numer. Meth. in Fluids.*, 17:323–348, 1993.

[22] J.E. Killough and M.F. Wheeler. Parallel iterative linear equation solvers: An investigation of domain decomposition solvers for reservoir simulation. In *Ninth SPE Symposium on Reservoir Simulation*. SPE paper no. 16021, San Antonio, Texas, 1987.

[23] H.M. Klie. *Krylov-secant methods for solving large-scale systems of coupled nonlinear parabolic equations*. PhD thesis, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, Sept. 1996.

[24] H.M. Klíe, L. Pavarino, M. Ramé, C. San Soucie, C. Dawson, and M.F. Wheeler. Preconditioners for Newton-Krylov methods for multiphase flows. In *Subsurface Modeling Group Industrial Affiliates Meeting*, August 1994.

[25] H.M. Klíe, M. Ramé, and M.F. Wheeler. Krylov–secant methods for solving systems of nonlinear equations. Technical Report TR95–27, Dept. of Computational and Applied Mathematics, Rice University, 1995.

[26] N.K. Nichols. On the convergence of two–stage iterative processes for solving linear equations. *SIAM J. Numer. Anal.*, 10:460–469, 1973.

[27] Y. Saad. A flexible inner–outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14:461–469, 1993.

[28] S. Turek. On discrete projection methods for the incompressible Navier–Stokes equations. In preparation, 1994.

[29] H.A. van der Vorst and C. Vuik. GMRESR: A Family of Nested GMRES Methods. Technical Report TR91–80, Technological University of Delft, 1991.

[30] J.R. Wallis. Incomplete gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *Seventh SPE Symposium on Numerical Reservoir Simulation*. SPE paper no. 12265, San Francisco, Texas, 1983.

[31] J.R. Wallis. Two-step preconditioning. Private Communication, 1993.

[32] A. Weiser and M.F. Wheeler. On convergence of block–centered finite differences for elliptic problems. *SIAM J. Numer. Anal.*, 25:351–375, 1988.

[33] J.A. Wheeler and R.A. Smith. Reservoir simulation on a hypercube. In *64th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers*. SPE paper no. 19804, San Antonio, Texas, 1989.