Two-Stage Preconditioners for Inexact Newton Methods in Multi-phase Reservoir Simulation

Héctor Klíe
Marcelo Ramé
Mary Wheeler

CRPC-TR96641-S
January 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

HÉCTOR KLÍE *, MARCELO RAMÉ † AND MARY F. WHEELER ‡

Abstract. Two-stage procedures refers to a family of convergent nested or inner-outer iterations. This paper addresses their use as preconditioners in the context of systems of coupled nonlinear partial differential equations, specifically those modeling underground multiphase flow phenomena. The linear systems arising after the discretization and the Newton linearization are highly nonsymmetric and indefinite but coefficient blocks associated with a particular type of unknown possess properties that can be exploited to enhance the overall conditioning of the coupled system. We show that decoupling strategies combined with two-stage preconditioners provide an efficient device to accelerate Krylov subspace methods such as GMRES and BiCGSTAB. Theoretical discussion and numerical experiments reveal the suitability of this approach and contrast it to fairly robust, standard ones which "blindly" precondition the entire coupled linear system.

Keywords: Two-stage methods, preconditioners, Krylov iterative solvers, block methods, coupled nonlinear partial differential equations, inexact Newton methods, multi-phase flow and transport, reservoir simulation.

AMS(MOS) subject classification: 3504, 35Q35, 35M10

1. Introduction.

The advent of increasing computing power has been the driving force for solving larger scientific and engineering problems. Consequently, new numerical algorithms have been coming forth with this computer technology sophistication. 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. The present research tries to respond to this reality in the context of reservoir simulation. The need for solving problems with at least one million gridblocks, and several unknowns per gridblock, has become one of the main challenges in the reservoir community. Therefore the conception of robust and efficient solvers plays an important role in the oil industry research. Major challenges arise 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 which are also known in the literature as nested or inner-outer procedures; see e.g., [3, 4, 14, 22, 28, 32, 43]. 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 [54]) and the subsequent Newton linearization of the coupled algebraic system of nonlinear equations. These linear systems (i.e., instances of Newton equations) are highly nonsymmetric and 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.)

* Department of Computational and Applied Mathematics, Rice University, Houston Texas 77251, USA; E-Mail: klie@rice.edu. Support of this author has been provided by Intevep S.A., Los Teques, Edo. Miranda, Venezuela.

† Department of Computational and Applied Mathematics, Rice University, Houston Texas 77251, USA; E-Mail: marcelo@rice.edu.

‡ Texas Institute for Computational and Applied Mathematics, University of Texas, Austin, Texas 78712, USA; E-Mail: mfw@ticam.utexas.edu

Despite the difficulty of these linear systems, there are certainly some "nice" properties associated to the coefficient blocks that affect each type of variable. Under mild conditions, which are regularly met at a modest time step size, each of these blocks are irreducible and diagonally dominant. Moreover, the strict diagonal dominance in some of these blocks leads to the M-matrix property. These isolated algebraic properties can be exploited so that better conditioning can be achieved in the entire coupled system. Moreover, devices leading to this desirable situation would also aid to weaken the coupling introduced by the off-diagonal blocks representing the crossed discretization of the nonlinear partial differential equations. We call these devices decoupling operators and use them as a preprocessing step to facilitate the effectiveness of two-stage preconditioners.

We remark that different solvers or preconditioning strategies can be used as intermediate steps within this two-stage preconditioners. In fact, the idea can be generalized to multi-stage methods [18]. We do not pursue this idea further here, though. We rather center our attention on those two-stage algorithms that arise naturally in block type of preconditioning: block Jacobi, block Gauss-Seidel and Schur complement based. We include in our analysis a combinative preconditioner originally proposed in [9] and later restated as an inexact procedure in [52]. The combinative method relies primarily upon the solution of a reduced pressure based system. In order to strengthen its robustness we propose an additive and multiplicative extension of this combinative preconditioner in terms of pressure and saturation residuals. We also aim these preconditioners at adding efficiency and robustness of two well known Krylov subspace iterative methods: GMRES and BiCGSTAB.

It is worth mentioning that ideas to sequentialize (i.e. remove part of the fully implicitness in time) have played an important role not only in the time discretization formulation of multi-phase flow and transport in porous media simulation but also in the solution of Navier-Stokes equations governing fluid dynamics problems. Sequential solution methods can be regarded as strategies to decouple the system by means of operator splitting or time-lagging some of the variables present in the physical model. Along this trend, we have the well known IMPES (IMplicit Pressures-Explicit Saturations) formulation in reservoir simulation (see, e.g., [5]) and, for Navier-Stokes problems, the segregated methods in CFD [34, 35]. Such strategies can certainly be inspiring to generate preconditioners for coupled linear systems already treated under the fully implicit scheme. This geneal idea motivates the work we present in this paper.

The authors have recently formulated and evaluated a Hybrid Krylov Secant (HKS) method for solving nonlinear sets of equations [38]. The method represents a blend of a inexact Newton and a secant method but the key point in HKS that motivates the present work is as follows. The first Newton step is obtained iteratively by a Krylov-subspace method inside which information on the eigenvalue spectrum of the matrix is generated. After convergence to a given (adjustable) tolerance, the task of finding the subsequent Newton steps is given to a much cheaper fixed point iteration, with relaxation parameters given by the eigenvalue spectrum found in the Krylov iteration. This scheme is further optimized by secant updates to the various linear operators involved.

The Krylov iteration to obtain the first Newton step of the HKS algorithm is necessary to generate (nearly optimal) relaxation parameters which allow the inexpensive iterative method to converge rapidly. However, a solution to a large system of linear

equations by a Krylov method can be costly unless a good preconditioner is found. In particular, knowledge on how to construct preconditioners for linear systems with multiple unknowns per discretization element (or, equivalently, gridblock) is sparse at best in the literature [52]. This work proposes and evaluates several preconditioning schemes for such systems in the context of Krylov-space iterative solvers.

The present work in conjunction with HKS methods constitutes a framework for a new family of solvers for fully implicit formulations of equations for flow and transport in porous media. Although, we stress that our experiences are applicable to more general settings where systems of coupled equations arises such in device circuit simulation [6], CFD and control problems.

This paper is organized as follows. We begin Section 2 with a presentation of the equations governing the multiphase flow in porous media. We then describe their discretization and the linearization by the Newton method. We include into the discussion a brief description of the GMRES and BICGSTAB algorithms. In Section 3, we analyze the structure of the linear system to be solved at every Newton step. Section 4 focuses the discussion on two different decoupling operators and their implications in clustering the eigenvalues of the original coupled system. Section 5 is devoted to discuss the philosophy behind the family of two-stage procedures and to describe those preconditioners that the authors consider most appropriate for the type of modeling problem addressed in this work. Technical discussion is supported and further illustrated through experiments in Section 6. We conclude this work with some final remarks and suggestions for further research on the subject in Section 7.

2. Description of the Problem. The paper concentrates the analysis on the equations for black-oil simulation which constitute the simplest way to realistically model multi-phase flow and transport in porous underground formations. To further simplify the presentation we only look at the two-phase model. Extensions to multiple unknowns per gridblock is readily evident.

2.1. Differential Equations. The basic equations for black-oil reservoir simulation consist of conservation equations for oil, gas and water. However, for simplicity, we limit the presentation to 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 [5] and [39]. 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

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

where $\mathbf{K}$ 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: $P_c(S_w) = P_n - P_w$.
- Relative permeabilities depend on both location and saturation.

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

The simulator used in the experiments presented in this work can accomodate problems from both the petroleum and the environmental engineering disciplines for it can specify general boundary conditions given by

$$\sigma \mathbf{u}_w \cdot \vec{n} + \nu P_w = h_w, \tag{3}$$

$$\sigma \mathbf{u}_n \cdot \vec{n} + \nu P_n = h_n, \tag{4}$$

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. (In reservoir engineering, the typical boundary conditions are of Neumann type for both the saturation and pressure unknowns. The resulting (possibly) rank deficient linear system is solved by choosing the bottom hole pressure at a given reservoir location.)

Frequently, the primary unknowns in the preceding system of parabolic equations are pressures and saturations of one phase or two different phases (see the discussion of [5] about other possible formulations.) The primary unknowns in our simulator are P_n and S_n. All other variables can then be computed explicitly based on these two.

In the case of slight compressibility, it can be shown that the system is of mixed parabolic-hyperbolic character, with one nonlinear parabolic equation in terms of pressure and one nonlinear convection-diffusion equation in terms of saturation [25]. 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 basically depend on saturations such as relative permeability and capillary pressure. The pressure equation degenerates into an elliptic equation for incompressibility of both phases (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 quasilinear hyperbolic equation.

2.2. Discretization. Nowadays, reservoir simulators rely on a variety of discretization schemes in time, ranging from the IMPES to the fully implicit formulations (see [5] and [39] for detailed discussions.) In between these two extremes, semi-implicit [42] and adaptive implicit discretizations have been proposed [30].

However, the fully implicit formulation offers the highest robustness among these possible alternatives in long term simulation. The main drawback of fully implicit methods resides in the solution of a large nonlinear system of equations. If Newton method is employed then several nonsymmetric and indefinite linear systems need to be solved at each time step.

In the context of the two-phase problem being discussed in this work, both pressure and saturation (degrees of freedom) unknowns 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 $(l = o, w)$ 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 $n+1$ time level; the subscripts i, j and k indicate the gridblock location. The fraction term is approximated through upstream weighting and the permeability is weighted harmonically in the direction of the flow to account for variations in gridblock sizes.

Discretization of the model equations (1)-(2) is performed by block-centered finite differences (or, equivalently, by lowest-order mixed finite elements) obeying a seven point stencil for pressures and saturations of both phases, thus giving rise in general to 28 different coefficients associated with a given interanl gridpoint location. The entire discretization leads to a system of nonlinear algebraic equations given by

$$\begin{aligned}
\Delta x_i \Delta y_j \Delta z_k \phi_{ijk}((\rho_l S_l)^{T+1}_{ijk} - (\rho_l S_l)^n_{ijk}) &= \Delta t^n \Delta x_i \Delta y_j \Delta z_k q^{T+1}_{l,ijk} \\
&+ \Delta t^n \Delta y_j \Delta z_k \Bigg\{ \lambda_{l,i+\frac{1}{2}} \frac{P_{l,i+1,jk} - P_{l,ijk}}{\Delta x_{i+1/2}} - \\
&- [\lambda_{l,i+\frac{1}{2}} \rho_l g] \frac{Z_{i+1,jk} - Z_{ijk}}{\Delta x_{i+1/2}} - \\
&- \lambda_{l,i-\frac{1}{2}} \frac{P_{l,ijk} - P_{n,i-1,jk}}{\Delta x_{i-1/2}} + \\
&+ [\lambda_{l,i-\frac{1}{2}} \rho_l g] \frac{Z_{ijk} - Z_{i-1,jk}}{\Delta x_{i-1/2}} \Bigg\}^{T+1}_{jk} \\
&+ \text{similar terms for the y and z directions,}
\end{aligned} \tag{5}$$

where $\Delta x_{i+1/2} = (x_{i+1} - x_i)/2$, i.e., the cell midpoint along the x direction. In a similar way, $\Delta y_{i+1/2}$ and $\Delta z_{i+1/2}$ are defined. Higher degree of discretization has been considered in the context of IMPES formulations [45]. Dawson *et al.* [21] consider a 19-point stencil in space within a fully implicit parallel reservoir simulator to handle underground heterogeneities. They use a full permeability tensor implementation together with general boundary condition specifications within each subdomain.

The extra relations mentioned in the previous subsection and their corresponding partial differentiation with respect the primary unknows are used as part of the Newton linearization of each of the nonlinear conservation equations. Some direct and indirect simplifications are performed, without affecting the validity of the numerical approximation, as result of small compressibility coefficients accompanying the linearized terms.

The above procedure follows the description by Wheeler and Smith [55] on developing a parallel black-oil simulator. Further insights about discretization of these equations can be found in [5] and [25].

2.3. Newton formulation. The fully implicit formulation for the numerical solution of systems of nonlinear parabolic equations leads us to solving the following nonlinear problem for each time step

$$F(u) = 0,$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Here, the vector u represents unknowns in pressures and saturations of one particular phase.

The composition of Newton with a Krylov iterative solver (such as GMRES or BiCGSTAB) with a criterion for defining linear tolerances dynamically, and a line-search backtracking strategy [24] is the basis of our inexact Newton algorithm. This algorithm is described as follows:

ALGORITHM 2.1.

1. Let u_0 be an initial guess.
2. For $n = 0, 1, 2, \ldots$ until convergence do
   2.1 Choose $\eta_k \in [0, 1)$.
   2.2 Compute a vector s_k satisfying

$$J(u_k)\, s_k = -F(u_k) + r_k,$$

   with $\frac{\|r_k\|}{\|F(u_k)\|} \leq \eta_k$, by some iterative method.
   2.3 Set $u_{n+1} = u_k + \lambda_k s_k$, where λ_k is the line search damping parameter.

The step length λ_k is computed using the linesearch backtracking scheme which ensures a decrease in $f(u) = \frac{1}{2} F(u)^t F(u)$. Step 2.2 should force s_k to be a descent direction for $f(u_k)$. That is,

$$\nabla f(u_k)^t s_k = F(u_k)^t J(u_k)\, s_k < 0,$$

in such case, we can assure that there is a ζ_0 such that $f(u_k + \zeta s_k) < f(u_k)$ for all $0 < \zeta < \zeta_0$. In practice, the final residual given by the iterative linear solver is acceptable whenever the Dembo-Eisenstat-Steihaug condition is met [22], i.e.,

$$\|r_k\| = \|F(u_k) + J(u_k)\, s_k\| < \eta \|F(u_k)\|, \quad 0 < \eta < 1. \tag{6}$$

Heuristics to select the linear tolerances or forcing terms, η_k, in Step 2.1 have been subject to extensively detailed research by Eisenstat and Walker [26, 27]. Practical experiences of their work in the context of reservoir simulation are reported in [21].

2.4. Iterative Method Framework. The typical problem sizes encountered in large-scale reservoir simulation rule out the use of direct methods to solve the linear systems arising in the Newton iteration. Consequently, inexact Newton methods are preferred. Although the theory of inexact Newton solvers is relatively recent [22], iterative methods such as SIP, SOR, CGS and ORTHOMIN have been of common practice in reservoir engineering for a number of years (see [39] for a general overview.) These four algorithms (and some others) have lost popularity over time on account of their lack of robustness in dealing with physical conditions common in reservoir engineering applications of current interest. Multigrid has been also investigated [23, 46] but its effectivity has only been shown for moderate rock heterogeneity and 2-D problems. Lately, Krylov subspace methods like BiCGSTAB, Chebyshev iterations

and GMRES have been employed as inner solvers for inexact Newton methods (see e.g., [33] and references therein.)

The use of Krylov subspace methods as inexact solvers within the Newton method was consolidated in [15]. Since then, extensions to the theory have been rapidly appearing in the literature and bringing insight about the capabilities of Krylov subspace methods (mainly GMRES) in the developing of globalization strategies [17, 19, 27].

Further description and several pointers into the literature of Krylov subspace iterative methods can be found in [7, 31]. Here, we just state some of the highlights of GMRES and BiCGSTAB.

Given a linear operator A in $\mathbb{R}^{n\times n}$ and a vector v in $\mathbb{R}^n$, the Krylov subspace $\mathcal{K}_n(A,v)$ is defined as

$$\mathcal{K}_n(A,v) \equiv span\{v, Av, A^2v, ..., A^{n-1}v\}.$$

There are basically two types of approaches for solving a given linear system $Ax = b$ by an iterative procedure defined in terms of a Krylov subspace. Let x_0 be a initial approximation towards the solution of this system, and $r_0 = b - Ax_0$ be the corresponding residual. We can either consider

- A minimal residual approximation: Choose $z_n \in \mathcal{K}_n(A,v)$ and solve

$$\min_{z\in\mathcal{K}_n(A,r_0)} \|b - A(x_0+z)\| = \min_{z\in\mathcal{K}_n(A,r_0)} \|r_0 - Az\|, \tag{7}$$

or,

- A Galerkin approximation: Choose $z_n \in \mathcal{K}_n(A,v)$ so that

$$r_n = r_0 - Az_n \perp \mathcal{K}_n(A,r_0). \tag{8}$$

Both formulations, find an approximate solution by setting $x_n = x_0 + z_n$. Here, $\|\cdot\|$ denotes the Euclidean norm.

The GMRES algorithm works under the philosophy suggested by (7) whereas BiCGSTAB follows (8). We now proceed to briefly describe each one.

2.4.1. GMRES. The GMRES algorithm [44] generates a basis for the Krylov space through the Arnoldi process. The fundamental point of this process is to create a decomposition that can be written as:

$$AV_n = V_nH_n + h_{n+1,n}v_{n+1}e_n^t,$$

or as

$$AV_n = V_{n+1}\bar{H}_n,$$

where

$$V_{n+1} = [V_n|v_{n+1}], \qquad \bar{H}_n = \begin{pmatrix} H_n \\ h_{n+1,n}e_n^t \end{pmatrix}.$$

The matrix V_n is orthogonal and its columns represent a basis for $\mathcal{K}_n(A,v)$ and $\bar{H}_n$, is an $(n+1)\times n$ upper Hessenberg matrix of full rank n. Hence, the minimal residual approximation can be rewritten as the following least squares problem

$$\min_{y\in\mathbb{R}^n} \left\| \|r_0\| e_1 - \bar{H}_n y \right\|.$$

One of the strongest arguments for using GMRES is its capability of producing monotonically decreasing residual norms. For a problem size n, the theory predicts that convergence is achieved within n iterations in the absence of roundoff errors. However, m iterations of GMRES requires $\mathcal{O}(m^2 n)$ operations and $\mathcal{O}(mn)$ of storage, making the procedure infeasible for large values of m. Restarting GMRES after m steps (with $m \ll n$) alleviates the problem but sacrifices its nice convergence properties. However, the restarted version of GMRES works well in practice specially with good preconditioning strategies.

2.4.2. Bi-CGSTAB. The BiCGSTAB algorithm [48] was developed to overcome the erratic converging behavior shown by the Conjugate Gradient Squared method (CGS) and the Bi-Conjugate Gradient method (Bi-CG.)

In the Bi-CGSTAB algorithm the iterates are constructed in such a way that the residual r_i is orthogonal with respect to a sequence of vectors $\{\widetilde{r}_i\}_0^{i-1}$ and in the same way, $\widetilde{r}_i$ is orthogonal to $\{\widetilde{r}_i\}_0^{i-1}$ (biorthogonality condition.) The i-th residual can be expressed as $r_i = P_i(A)\, r_0$, where P_i is a monic polynomial of degree less or equal to i. The $\widetilde{r}_i$ are generated with polynomials of the form $Q_i(x) = \prod_{j=1}^{i}(1-\omega_j x)$, where the ω_j are chosen so that

$$\left(P_i(A)\, r_0, Q_j\left(A^t\right)\widetilde{r}_0\right) = 0, \qquad\qquad i \neq j.$$

This last condition is enforced without an explicit reference to A^t (as it is done in CGS.) BiCGSTAB has short recurrences, requires only two A products per iteration and produces a solution $x_k \in x_0 + \mathcal{K}_{2k}(A, r_0)$. It typically produces much smoother residual norm behavior than CGS, but the residuals norms still behave badly in some problems, specially in discretized diffusion-advection equations with dominant advection components. Some improvements to the algorithm are discussed and referred to in [7].

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 2.1. We identify properties associated with the blocks conforming 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 produce convergence of the Newton method and, therefore, provide a framework for evaluating the last advances in preconditioning coupled 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

$$Jx = f \Leftrightarrow \begin{pmatrix} J_{pp} & J_{ps} \\ J_{sp} & J_{ss} \end{pmatrix} \begin{pmatrix} p \\ s \end{pmatrix} = - \begin{pmatrix} f_n \\ f_w \end{pmatrix}. \tag{9}$$

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

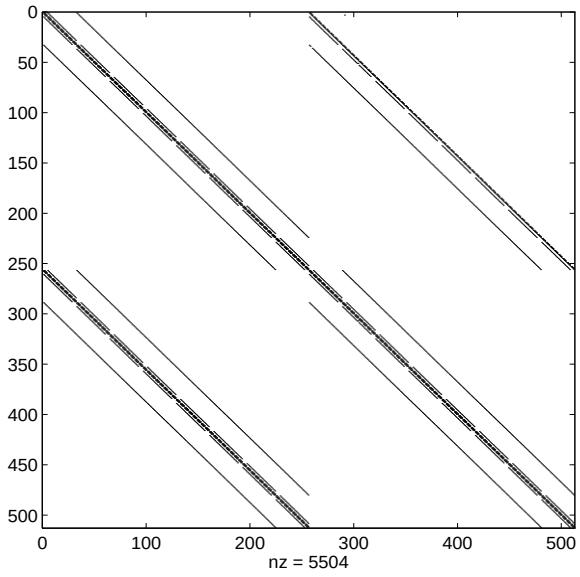

FIG. 1. *Matrix structure of linear systems in the Newton iteration.*

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

The block J_{pp}, containing pressure coefficients, has the structure of a purely elliptic problem in the the nonwetting phase pressures. The block J_{ps} of the Jacobian matrix has a structure similar to that of a discretized first-order hyperbolic problem in the nonwetting phase saturations. J_{sp} has the coefficients of a convection-free parabolic problem in the nonwetting phase pressure and, finally, J_{ss} represents a parabolic (convective-diffusive) problem in the oil saturations.

The position of nonzero entries of a given Jacobian matrix is shown in Figure 1. In this particular example, we can observe the effect of the upstream weighting within the block J_{ps}: the moving front is one block behind giving the only nonzero coefficients in the lower part of the block. However, the absent values in the upper part are added positively 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 (further discussion about this issue is given in [5].) In general, in system (9), the block coefficients J_{pp}, J_{ps} and J_{sp} the following properties (see e.g., [2, 5] for further physical insights and [3, 10] 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_{ps} and J_{sp} as result of compressibility terms and pore factors contribution into the main diagonal of these blocks. In consequence, these blocks are nonsingular, positive stable and M-matrices. Strict diagonal dominance for some of the the rows of J_{pp} can be achieved by the contribution of bottom hole pressures specified as part of the boundary conditions.

In this case, this block is an irreducibly diagonally dominant matrix. In addition, under small changes of formation of volume factors and flow rates between adjacent gridblocks we can expect both blocks J_{pp} and J_{sp} be nearly symmetric.

The saturation coefficient block $-J_{ss}$ presents algebraic properties similar to the other blocks. It has a convection-diffusion behavior characterized by capillary pressure derivative terms (the diffuse part) and wetting relative permeability derivative terms (the convective part.) The diffusive part becomes dominant over the convective part when capillary pressure gradients are higher with respect relative permeabilities gradients of the wetting phase. It is likely that this occurs at the beginning and end of the of the simulation when the capillary pressure curve tends to be steeper. During intermediate time steps of the simulation, the wetting pressure gradients and relative permeabilities gradients with respect wetting saturations are less pronounced and affect negatively the magnitude of the convective part. However, under the same trend the capillary pressure derivatives with respect wetting saturations are less prominent affecting negatively the amount of dispersivity.

Desirable diagonal dominance in $-J_{ss}$ can be indeed 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 but also the Newton method itself suffers to converge even at steps obtained by a fair solution of the linear system. Hence, it is our opinion that the conditioning of this block is crucial in the conditioning of the entire coupled system. The reader can verify the resemblance between the spectrum of J_{ss} and the Jacobian matrix J through inspection of Fig. 2 and Fig. 3.

We should stress that the "degree" of diagonal dominance is proportional to the pore volume of the gridblocks and inversely proportional to the time step size. On the other hand, definition of bottom hole pressures as part of the boundary conditions affects positively the diagonal dominance of the blocks, whereas specified rates in the source wells affects the diagonal dominance in a negative way.

In this work, we assume the blocks J_{pp} and $-J_{ss}$ being irreducibly diagonally dominant and the blocks J_{ps} and J_{sp} being diagonally dominant. Taking into account the minus sign in front of J_{ss} all blocks are Z-matrices with positive diagonal entries.

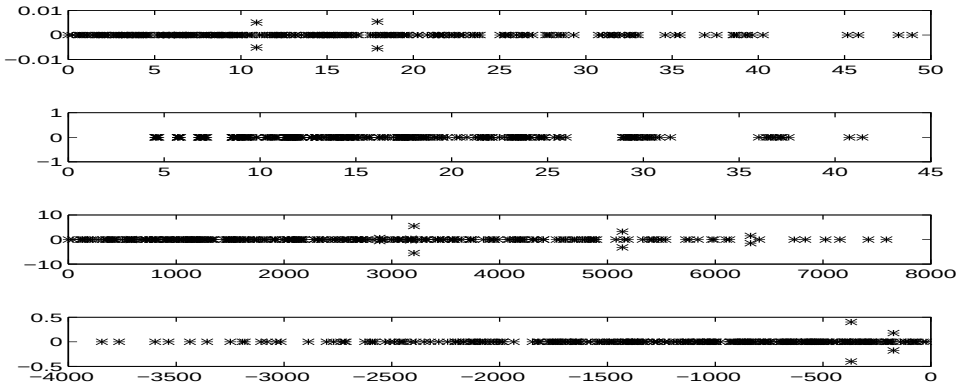

FIG. 2. *Spectra of the blocks composing the sample Jacobian matrix. From top to bottom, they correspond to* J_{pp}, J_{ps}, J_{sp} *and* J_{ss}.

It is clear that these conditions do not guarantee nice properties on the whole matrix J. Moreover, the Jacobian matrix is highly nonsymmetric and indefinite in principle. This is the main argument in favor of decoupling strategies to generate preconditioners for (9) since we can exploit better convergence properties out of the blocks than from the complete system itself.

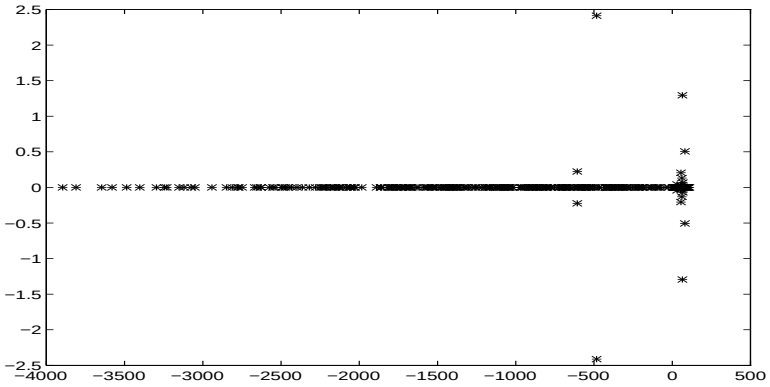

FIG. 3. *Spectrum of the sample Jacobian matrix to be used throughout the discussion on two-stage preconditioners.*

In the forthcoming section, we progressively illustrate our analysis by looking at spectrum changes of a typical Jacobian matrix after applying different operators. In this particular case, we consider a Jacobian matrix resulting from a small scale reservoir simulation (i.e. a grid problem size of $8 \times 8 \times 4$) where the blocks J_{pp}, J_{ps}, J_{sp} and $-J_{ss}$ are positive stable, as clearly depicted in Fig. 2. From Fig. 3 we can infer 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. In this section we describe the role that decoupling operators play in the definition of robust and efficient two-stage preconditioners. The basic goal is to weaken, as a preprocessing step, the coupling represented by offdiagonal blocks within the coupled system.

The idea of decoupling operators has been barely stated not only in the general literature but also implicitly treated in works on solvers for reservoir simulation. 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 saturations likely occurring at high flow rates or at larger time steps.

Bank *et al.* [6], with their alternate-block factorization (ABF) method, propose a simple way to weaken the coupling of system drift-diffusion equations that occur in semiconductor device modeling. Under the light of highly simplifying assumptions, however, they analyze the viability of the decoupling process for preconditioning linear systems.

Their work turns out to be of value in the context of multiphase flow since their decoupling operator leads to a significant clustering of eigenvalues associated with Jacobian matrices ocurring during the simulation process. Moreover, in very rare cases (detected only after extensive experimentation with synthetic random matrices

whose blocks obey our assumptions for the blocks of J), the resulting decoupled system fails to have all eigenvalues lying at 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.

We then proceed to characterize the decoupling operators and their implications in preconditioning the coupled system.

4.1. Block decoupling. Consider the Jacobian system shown in (9) and let us define

$$D = \begin{pmatrix} D_{pp} & D_{ps} \\ D_{sp} & D_{ss} \end{pmatrix} = \begin{pmatrix} diag(J_{pp}) & diag(J_{ps}) \\ diag(J_{sp}) & diag(J_{ss}) \end{pmatrix}, \tag{10}$$

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

$$\begin{aligned} J^D \equiv D^{-1}J &= \begin{pmatrix} \Delta^{-1} & 0 \\ 0 & \Delta^{-1} \end{pmatrix} \begin{pmatrix} D_{ss}J_{pp} - D_{ps}J_{sp} & D_{ss}J_{ps} - D_{ps}J_{ss} \\ D_{pp}J_{sp} - D_{sp}J_{pp} & D_{pp}J_{ss} - D_{sp}J_{ps} \end{pmatrix} \\ &\equiv \begin{pmatrix} J^D_{pp} & J^D_{ps} \\ J^D_{sp} & J^D_{ss} \end{pmatrix}, \end{aligned} \tag{11}$$

where $\Delta \equiv D_{pp}D_{ss} - D_{ps}D_{sp}$, and the superscript D has been introduced for later notational convenience. We can observe that the main diagonal of the main diagonal blocks is equal to one. Conversely, the main diagonal entries of the offdiagonal 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.* [6] observe that this operation weakens the coupling between the partial differential equations which turns out to be in our particular case, the equation for each phase.

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 saturations derivatives were neglected from the transmissibilities components. 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 loose the inherent diagonal dominance of J. Other implications of this choice will be discussed in the next subsection.

The above decoupling or ABF operator admits an alternate representation. We can associate smaller matrix blocks with with individual unknowns within the mesh. This means to permute rows and columns in an interleaved fashion and to number every pressure unknown followed by the saturation unknown at the same gridblock and repeat this for every gridblock. Let P be the matrix that performs such permutation

and define

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

where

$$\widetilde{J}_{i,j} = \begin{pmatrix} (J_{pp})_{i,j} & (J_{ps})_{i,j} \\ (J_{sp})_{i,j} & (J_{ss})_{i,j} \end{pmatrix},$$

is the 2×2 matrix representing the coupling between unknowns.

It clearly follows for an invertible D that

$$\widetilde{D} = PDP^t \Leftrightarrow \widetilde{D}^{-1} = PD^{-1}P^t.$$

Hence, $\widetilde{D}^{-1}$ is a block diagonal matrix given whose blocks are the inverse of each local problem at each gridblock. That is,

$$\widetilde{D}^{-1} = \begin{pmatrix} \widetilde{J}_{1,1}^{-1} & 0 & \cdots & 0 \\ 0 & \widetilde{J}_{2,2}^{-1} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \widetilde{J}_{nb,nb}^{-1} \end{pmatrix}. \tag{12}$$

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 [9] comment about the possibility of decoupling more equations in their combinative method but only with respect pressure coefficients. They did not forsee the positive effect, as we shall see here, that a full decoupling of the gridblock has in conditioning the system.

The core of the combinative approach is the effective solution of pressure based systems. In this situation, there is no need to go beyond in the decoupling process as expressed in (12). The coefficients introducing the coupling with pressures are eliminated within the gridblock by Gauss elimination so that corresponding coefficients at neighboring gridblocks are expected to become small. To be more precise, let

$$\widetilde{W}_p = \begin{pmatrix} (\widetilde{W}_p)_1 & 0 & \cdots & 0 \\ 0 & (\widetilde{W}_p)_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & (\widetilde{W}_p)_{nb} \end{pmatrix}, \tag{13}$$

where

$$(\widetilde{W}_p)_i = I_{nu\times nu} - e_1 e_1^t + \left(e_1^t J_{ii} e_1\right) e_1 e_1^t \widetilde{J}_{ii}^{-1},$$

and $e_1 = (1,0)^t$. Therefore, the operator $\widetilde{W}_p$ is a block diagonal matrix that removes the coupling in each 2×2 diagonal block with respect to the pressure unknown. In fact, it readily follows that

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

Similarly, we could define an operator $\widetilde{W}_s$ with the canonical vector $e_2 = (0,1)^t$. The operator $\widetilde{W}_p$ was introduced by Wallis in his IMPES two-stage preconditioner [52].

The consecutive counterpart, W_p, of the alternate representation of operator $\widetilde{W}_p$ is given by

$$
\begin{aligned}
J^{W_p} \equiv W_p J &= \begin{pmatrix} \Delta^{-1} D_{pp} & 0 \\ 0 & I_{nb\times nb} \end{pmatrix} \begin{pmatrix} D_{ss}J_{pp} - D_{ps}J_{sp} & D_{ss}J_{ps} - D_{ps}J_{ss} \\ J_{sp} & J_{ss} \end{pmatrix} \\
&\equiv \begin{pmatrix} J_{pp}^{W_p} & J_{ps}^{W_p} \\ J_{sp}^{W_p} & J_{ss}^{W_p} \end{pmatrix}.
\end{aligned} \tag{14}
$$

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

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

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

for $k = 1, 2, \ldots, nb$. In this particular lexicographic alternate ordering of unknowns, we could also define $j = 2 + 2\,(k-1)$ for R_s^t in order to obtain the corresponding saturation coefficients.

Finally, we stress that this presentation can be easily extended to more unknowns sharing a given gridpoint (e.g., three phases and multi components systems.)

4.2. Properties of the partially decoupled blocks. In general, it is a difficult task (and in fact, an open problem in many related fields [1, 6, 12, 29, 40]) to characterize properties associated with the entire 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 are based on some specific assumptions on the matrix J. For the class of matrices that we obtain, there is not yet an easy-to-check theoretical result that determines when a matrix is positive stable and moreover, when the symmetric part of a matrix 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. Specially, one would like to know if the eigenvalues are located in the right half plane of the complex plane to guarantee that convergence theory of the iterative method is valid. Also important is to detect a possible clustering of the eigenvalues since this may increase the rate of convergence. In this section, we briefly present two immediate results related to the individual diagonal blocks composing the already partially decoupled Jacobian matrix through D. Consider the decoupled matrix with a block-partitioned representation as showed in (9).

THEOREM 4.1. *Let J_{pp} and $-J_{ss}$ be diagonally irreducibly Z-matrices and let J_{ps} and J_{sp} be M-matrices in $R^{nb\times nb}$, then J_{pp}^D and J_{ss}^D are M-matrices.*

Proof. We prove separately that J_{pp}^D and J_{ss}^D are Z-matrices and strictly diagonally dominant matrices. Then by [3, Lemma 6.3, page 204] it immediately follows that they

are M-matrices. We only show that J_{pp}^D is a M-matrix. The proof for J_{ss}^D proceeds similarly.

First, note that $(\Delta^{-1})_{i,i} < 0, \forall i = 1, 2, \ldots nb$. In fact, $(D_{pp})_{i,i}, (D_{ps})_{i,i}$ and $(D_{sp})_{i,i}$ are all positive and $(D_{pp})_{i,i}$ is negative for $i = 1, 2, \ldots nb$, so that

$$\left(\Delta^{-1}\right)_{i,i} = (D_{pp}D_{ss} - D_{sp}D_{ps})_{i,i} = (D_{pp})_{i,i}(D_{ss})_{i,i} - (D_{sp})_{i,i}(D_{ps})_{i,i} < 0.$$

Therefore

$$\left(J_{pp}^D\right)_{i,j} = \left(\Delta^{-1}\right)_{i,i}\left[(D_{ss})_{i,i}(J_{pp})_{i,,j} - (D_{ps})_{i,i}(J_{sp})_{i,j}\right] \geq 0, \forall i \neq j, i, j = 1, 2, \ldots nb$$

since $(J_{pp})_{i,,j} \leq 0$ and $(J_{sp})_{i,j} \leq 0, \forall i \neq j$, $i, j = 1, 2, \ldots nb$.

For the strict diagonal dominance, consider the summation over the absolute value of the elements (except the one lying on the main diagonal) along the $i-$th row of J_{pp}^D, then

$$\begin{aligned}
\sum_{\substack{j=1\\ j\neq i}}^{nb}\left|(J_{pp})_{i,j}\right| &= \sum_{\substack{j=1\\ j\neq i}}^{nb}\left|\left(\Delta^{-1}\right)_{i,i}(D_{ss}J_{pp} - D_{ps}J_{sp})_{i,j}\right| \\
&= \left|\left(\Delta^{-1}\right)_{i,i}\right|\left[\left|(D_{ss})_{i,i}\right|\sum_{\substack{j=1\\ j\neq i}}^{nb}\left|(J_{pp})_{i,j}\right| + \left|(D_{ps})_{i,i}\right|\sum_{\substack{j=1\\ j\neq i}}^{nb}\left|(J_{sp})_{i,j}\right|\right] \\
&< \left|\left(\Delta^{-1}\right)_{i,i}\right|\left[\left|(D_{ss})_{i,i}\right|\left|(D_{pp})_{i,i}\right| + \left|(D_{ps})_{i,i}\right|\left|(D_{sp})_{i,i}\right|\right] \\
&= 1.
\end{aligned}$$

∎

The inequality is obtained due to the strict inequality held in every row of J_{sp} (Note that we can not affirm this with the exclusive contribution from the irreducibly diagonal dominance in J_{pp}.) We can say then, that all entries of the transformed blocks are bounded by 1, which is the value that all entries have in the main diagonal.

COROLLARY 4.1. *The diagonal blocks J_{pp}^D and J_{ss}^D are positive stable.*

Proof. This is just the result stated in [3, Theorem 6.12,page 211]. ∎

In Fig. 4 we show the spectrum of the resulting Jacobian matrix after applying the decoupling operators D^{-1} and $\widetilde{W}$. Interestingly enough, the Jacobian spectrum has been significantly compressed and shifted to the right half of the complex plane with D^{-1}. In contrast, strategies that intent to preserve much of the original structure of the matrix perform very poorly as preconditioners (see the great resemblance between the spectrum of WJ and J.) Several experiments like this one have indicated that the best strategy is to break as much as possible the coupling between equations (or unknows) than trying to preserve some desirable properties of the individual blocks.

5. Two-stage preconditioners. We begin by giving a brief background as motivation for the forthcoming ideas. The order of the following presentation obeys roughly a chronology of how the ideas that led to the formulation of the various two-stage preconditioners arose. We discuss in detail Wallis two-stage preconditioner format [52], and a couple of extensions to it in additive and multiplicative form. We end this section with a more efficient approach consisting of the combination of the decoupling operator D^{-1} and the inexact solution of standard block preconditioners such as block Jacobi, block Gauss Seidel and Schur complement based.

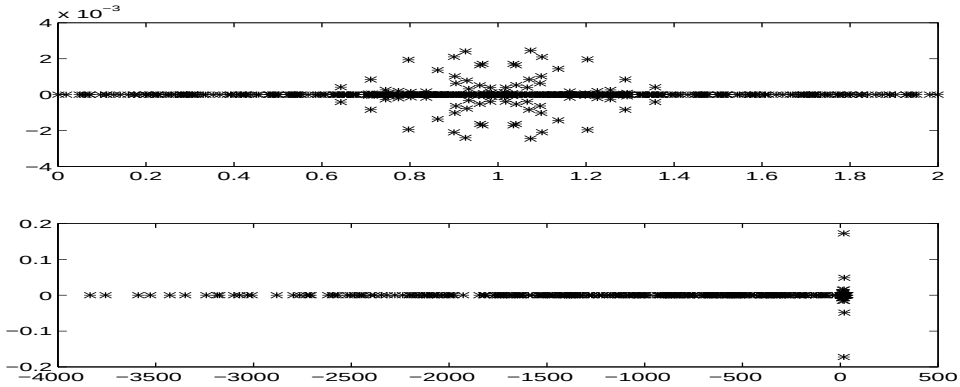

FIG. 4. *Spectra of the partially decoupled forms of the sample Jacobian matrix. The one above correspond to* $D^{-1}J$*, and the one below to* $\widetilde{W}\widetilde{J}$ *(or equivalently,* WJ*.)*

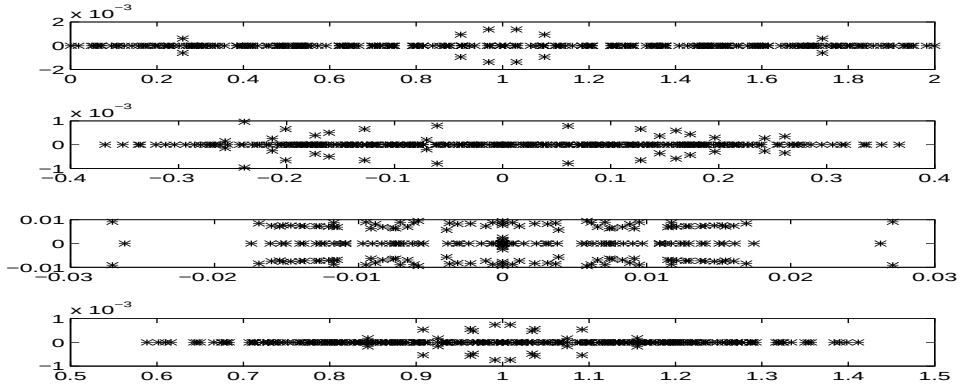

FIG. 5. *Spectra of each block after decoupling with* D*. From top to bottom, they correspond to the (1,1), (1,2), (2,1) and (2,2) blocks*

5.1. Background. Efforts to develop general and efficient solvers for systems coupled elliptic and parabolic equations have started to emerge strongly in the last years. However, Behie and Vinsome [9] appear to be the first to consider *combinative* preconditioners as a form of decoupled preconditioners in reservoir engineering. A minor change to the idea but seeking to incorporate saturation information was later proposed by Behie and Forsyth [8]. Wallis refined the original algebraic presentation of these authors and proposed the iterative solution of the pressures in order to tackle larger reservoir simulation problems [52]. Meanwhile, developments on the concatenation or combination of inexact preconditioning stages have been proposed for general symmetric and nonsymmetric problems [51], but specially in the context of domain decomposition [11, 13, 36] for flow in porous media. These works, however, do not address the the topic of specialized preconditioners for coupled equations.

In CFD the idea of using decoupled matrix blocks for the construction of preconditioners for iterative methods and for the implementation of solvers has been around longer. Segregated algorithms have been successfully applied for solving Navier-Stokes equations (see e.g., [35, 47] and references therein.) These methods rely upon the alternate solution of pressures and velocities or in the exhaustive solution of one of them to get a good overall solution of the problem. Similar type of ideas has been developed in sequential formulations at the level of time discretization rather on the level of linear solvers or preconditioners for fully implicit formulation.

The use of two-stage methods is not new (see e.g., [41] and references cited there). In fact, these methods are known under different names and are scattered throughout the literature. They are also known as inner-outer or inexact iterations [22, 28, 32]. In the context of preconditioning they have been referred to as nonlinear, variable or inner-outer preconditioners [3, 4, 43]. They have been also subject of study in parallel computing settings (e.g. see [18] 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 obeys primarily to the computational cost associated with solving large inner linear systems. Recent developments in Krylov-subspace methods have also contributed to the renewed interest in this area. For example, the Uzawa algorithm has been around for more than 35 years and it was recently that some researchers formalized the inexact version of this algorithm [14, 28]. In same fashion, intensive work has been devoted to extending current nonsymmetric iterative solvers to be able to accommodate the inexactness or variability of the preconditioner from iteration to iteration; e.g. [3, 43, 49, 50].

In this work, we use right preconditioning. It is well known that this form is preferable over left preconditioning for comparing different preconditioners since it makes the relative residual norms measured within the iterative solver invariant. This norm size invariance simplifies the implementation of globalization methods, such as linearch backtracking, within an inexact Newton procedure such as the one depicted in Algorithm 2.1. Furthermore, there is yet a more compelling reason for our adopting right preconditioning. If a left preconditioner is inexact or unavailable in closed form, there is no way to measure norms accurately, or rather consistently, throughout the various steps of an iterative method.

We only make an exception when we include the decoupling operator D^{-1} as a preprocessing step for the consecutive ordering of unknowns. In view that this operator is fixed, cheap and that its proper application introduces the desirable diagonal

dominance of the main diagonal blocks of the coupled system (9), we consider its use on the left. Hence, the application of D^{-1} implies the use of weighted norms for all vector norms. That is, if $r = (r_n, r_w)^t$ is a given residual (which concatenates residuals of both the wetting and the nonwetting phases) whose norm needs to be computed then

$$\|r\|_{D^{-1}} = \left\|\Delta^{-1}\right\| \left(\|D_{ss}r_n - D_{ps}r_w\|^2 + \|-D_{sp}r_n + D_{pp}r_w\|^2\right)^{\frac{1}{2}}.$$

Clearly, this does not introduce any major complication or overhead into the implementation. Moreover, this step can be also regarded as a scaling step for the coupled variables of the nonlinear function in a given Newton step. This incidentally improves the robustness of the whole Newton method. Further discussion on scaling within the Newton method can be seen in [24] and [16].

5.2. Combinative two-stage preconditioner. Consider the two-step 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 5.1.

1. Solve the reduced pressure system $\left(R_p^t \widetilde{J}^{W_p} R_p\right) p = R_p^t \widetilde{W}_p r$ 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

$$v = M^{-1}r = \hat{M}^{-1}\left[I - \left(\widetilde{J} - \hat{M}\right) R_p \left(R_p^t \widetilde{J}^{W_p} R_p\right)^{-1} R_p^t \widetilde{W}_p\right] r. \tag{15}$$

The preconditioner $\hat{M}$ is to be preferably computed once for each Newton iteration. This means that $\hat{M}$ should be easily factored. The system $\left(R_p^t \widetilde{J}^{W_p} R_p\right) p = R_p^t \widetilde{W}_p r$ is solved iteratively giving rise to a nested-like procedure. We finally remark that M_p is an exact left inverse of $\widetilde{J}$ on the subspace spanned by the columns of R_p. That is, $\left(M_p^{-1}\widetilde{J}\right) R_p = R_p$.

This is the preconditioner as stated by Wallis [52]. In contrast to the combinative method of Behie and Vinsome [9], 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 more appropriate the term two-stage combinative preconditioner according to a more accepted terminology for convergent nested inexact procedures and to the former designation employed by Behie and Vinsome. Fig. 6 shows the spectrum of the operator for an exact solution of the pressure system. In this particular example, $\hat{M}$ was taken to be the tridiagonal part of $\widetilde{J}$.

5.3. Additive and multiplicative extensions. With the use of $\widetilde{D}^{-1}$ and incorporating solution from saturations we can improve the quality of the previous preconditioner. We propose two different ways to accomplish this: additively and

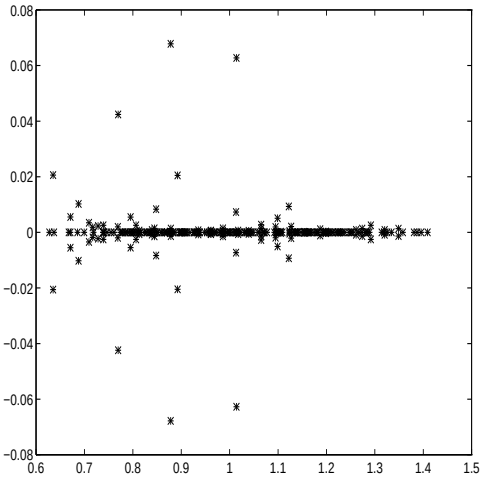

FIG. 6. *Spectra of the Jacobian right-preconditioned by the exact version of the combinative operator.*

multiplicatively. In the following we present both procedures for computing the preconditioned residual $v = M_{\text{add}}^{-1} r$ and $v = M_{\text{mult}}^{-1} r$. The additive combinative two-stage preconditioner is given by

ALGORITHM 5.2.

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

The multiplicative combinative two-stage preconditioner proposes the sequential treatment of the partially preconditioned residuals instead. In algorithmic terms it is given by

ALGORITHM 5.3.

1. Solve the reduced pressure system $\left(R_p^t \widetilde{J}^D R_p\right) p = R_p^t \widetilde{D}^{-1} r$ and denote its solution by $\hat{p}$.
2. Obtain expanded solutions $p = R_p \hat{p}$.
3. Construct new residuals $\hat{r} = r - J p$.
4. Solve the reduced saturation system $\left(R_s^t \widetilde{J}^D R_s\right) s = R_s^t \widetilde{D}^{-1} \hat{r}$ and denote its solution by $\hat{s}$.
5. Obtain expanded solutions $s = R_s \hat{s}$.
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 saturations are solved exactly and introducing the notation

$$t_l \equiv R_l \left(R_l^t \widetilde{J}^D R_l\right)^{-1} R_l^t \widetilde{D}^{-1}, \tag{16}$$

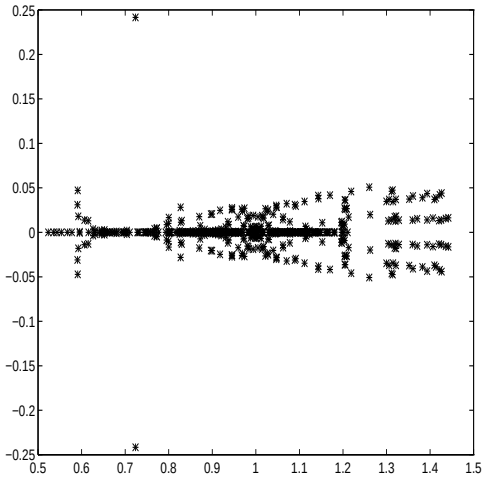

FIG. 7. *Spectra of the Jacobian right-preconditioned by the exact version of the two-stage additive operator.*

for $l = p, s$, the action of these preconditioners can be characterized by

$$v = M_{\text{add}}^{-1} r = \hat{M}^{-1} \left[I - \left(\tilde{J} - \hat{M} \right) (t_p + t_s) \right] r, \tag{17}$$

and

$$v = M_{\text{mult}}^{-1} r = \hat{M}^{-1} \left[I - \left(\tilde{J} - \hat{M} \right) \left(t_p + t_s - t_s \tilde{J}^D t_p \right) \right]. \tag{18}$$

The difference between the two preconditioners resides in the inclusion of the cross term $t_s \tilde{J}^D t_p$ resulting from the computation of a new residual in Step 3 of Algorithm 5.3. This residual is next improved by saturation solutions. Preliminary computational insights of these preconditioners were presented in [37]. In Fig. 7 and Fig. 8 we can observe the job that these preconditioners do 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 even though the resulting preconditioned system has a negative eigenvalue.

5.4. Block two-stage preconditioners. In the same way that decoupling operators have interpretation in block or alternate form, we can express the preconditioners described above in block form. However, in this opportunity we present them in a simpler form given that the decoupling operator performs a "good" job in 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 $\hat{M}$ as it is depicted in the combinative preconditioner and its corresponding additive and multiplicative extensions. The reason for this is that the overhead introduced by this operation is difficult to compensate for by its own preconditioning effectivity. In Section 6 we extend the analysis of its action to reinforce this view.

To facilitate the presentation we consider the factored form of the block-partitioned system (11),

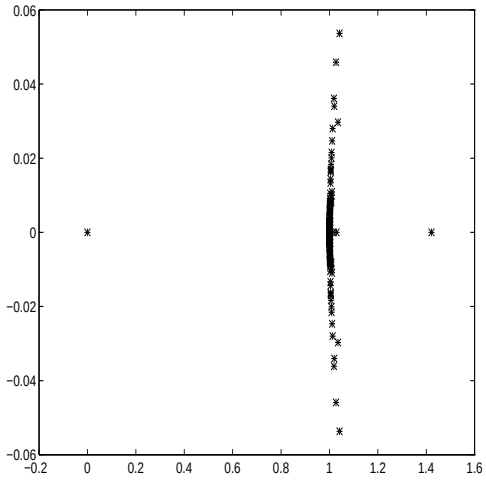

FIG. 8. *Spectra of the Jacobian right-preconditioned by the exact version of the two-stage multiplicative operator.*

$$J^D = \begin{pmatrix} I_{nb\times nb} & J^D_{ps}\left(J^D_{ss}\right)^{-1} \\ 0 & I_{nb\times nb} \end{pmatrix} \begin{pmatrix} S^D & 0 \\ 0 & J^D_{ss} \end{pmatrix} \begin{pmatrix} I_{nb\times nb} & 0 \\ \left(J^D_{ss}\right)^{-1} J^D_{sp} & I_{nb\times nb} \end{pmatrix}, \tag{19}$$

so that

$$\begin{aligned} \left(J^D\right)^{-1} = & \begin{pmatrix} \left(S^D\right)^{-1} & 0 \\ -\left(J^D_{ss}\right)^{-1} J^D_{sp}\left(S^D\right)^{-1} & \left(J^D_{ss}\right)^{-1} \end{pmatrix} \times \\ & \begin{pmatrix} I_{nb\times nb} & -J^D_{ps}\left(J^D_{ss}\right)^{-1} \\ 0 & I_{nb\times nb} \end{pmatrix}, \end{aligned} \tag{20}$$

where $S^D = J^D_{pp} - J^D_{ps}\left(J^D_{ss}\right)^{-1} J^D_{sp}$ is the Schur complement of J^D with respect J^D_{ss}.

Therefore, 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 (20) can be described as follows.

ALGORITHM 5.4.

1. Solve $J^D_{pp} q = r^D_n$ and denote its solution by $\widehat{q}$.
2. $w = -J^D_{sp}\widehat{q} + r^D_w$.
3. Solve $S^D s = w$ and denote its solution by $\widehat{s}$.
4. $y = J^D_{ps}\widehat{s}$.
5. Solve $J^D_{pp} z = y$ and denote its solution by $\widehat{z}$.
6. $\widehat{p} = \widehat{q} - \widehat{z}$.
7. Return $(\widehat{p}, \widehat{s})$.

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.

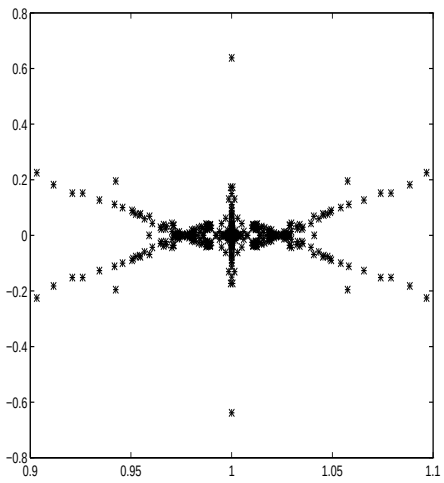

FIG. 9. *Spectra of the Jacobian right-preconditioned by the exact version of the two-stage block Jacobi operator.*

Clearly a preconditioner like this is costly to implement in our context. However, under this presentation it is straightforward to devise the steps for carrying out the action of different approximations to $(J^D)^{-1}$. When J_{ps} and J_{sp} are assumed to be zero matrices we obtain the two-stage block Jacobi (2SBJ) preconditioner. The two-stage Gauss-Seidel (2SGS)results from neglecting only the block J_{sp}. A more robust preconditioner can be obtained by means of a better approximation of the Schur complement, S, where all blocks of the original matrix are involved. In order to maintain this approximation under reasonable computational costs, it is customary to provide a simple approximation to $(J^D_{pp})^{-1}$.

Spectrum of these preconditioners for exact solution of the block subsystems are shown in Fig. 9-11. In Fig. 9, we can observe the significant clustering of the eigenvalues around the complex point $(1, 0)$ produced by 2SBJ preconditioning. Not surprisingly, the 2SGS block preconditioner does an even better job of clustering the eigenvalues except for one that appears separated from the rest as shown on Fig. 10. There is also a certain resemblance between the action of this preconditioner and that of the multiplicative two-stage preconditioner although the latter leaves one eigenvalue on the left half of the complex plane. This fact illustrates the close relationship between these preconditioners which shall become more evident in the next section.

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 [35]) ranging from solving separately for each nodal unknown to solving simultaneously for all the degrees of freedom associated with one or more (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 [47] to solve saddle point formulations arising from the discretization of Navier-Stokes equations for the incompressible flow. The algorithm departs from an approximation to the Schur com-

plement with respect pressures and solves iteratively the hyperbolic component given by the velocities (role represented by saturations in our case). That is, to obtain the preconditioned residual (r_n, r_w) we perform the following steps

ALGORITHM 5.5.

1. Set $\left(\bar{J}_{ss}^D\right)^{-1} \simeq (J_{ss}^D)^{-1}$.
2. Solve $\left[J_{pp}^D - J_{ps}^D \left(\bar{J}_{ss}^D\right)^{-1} J_{sp}^D\right] v_p = r_n^D - J_{ps}^D \left(\bar{J}_{ss}^D\right)^{-1} r_w^D$ iteratively. Obtain $\hat{v}_p$.
3. Solve $J_{ss}^D v_s = r_w^D - J_{sp}^D \hat{v}_p$ iteratively. Obtain $\hat{v}_s$.
4. Return $(\hat{v}_p, \hat{v}_s)^t$, 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 saturations. Note that this presentation comes from a different factorization to J^D given by (19). Conversely, the Schur complement with respect pressures leads to emphasizing more the saturation components. Thus, in agreement to the IMPES method we consider Algorithm 5.4 to give a more sound physical and numerical alternative. From now on, we refer to this preconditioner as discrete-projection two-stage preconditioner.

The operator $\widetilde{J}_{ss}^{-1}$is chosen to be computationally cheap. Turek [47] suggests $\widetilde{J}_{ss}^{-1}$ to be the inverse of the diagonal part of J_{ss} (i.e. Jacobi preconditioner.)

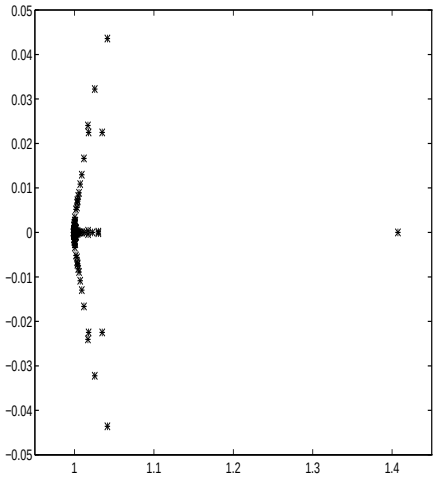

FIG. 10. *Spectra of the Jacobian right-preconditioned by the exact version of the two-stage block Gauss-Seidel operator.*

5.5. Relation between alternate and consecutive forms. If M indicates any of the preconditioners described above for the Jacobian matrix J, it is it is desirable that

$$\left\| I - JM^{-1} \right\| \leq \sigma < 1 \tag{21}$$

which implies the following two conditions

- Coercivity:

$$\left\langle JM^{-1}x, x \right\rangle \geq (1-\sigma)\langle x, x \rangle, \quad \forall x \in R^n.$$

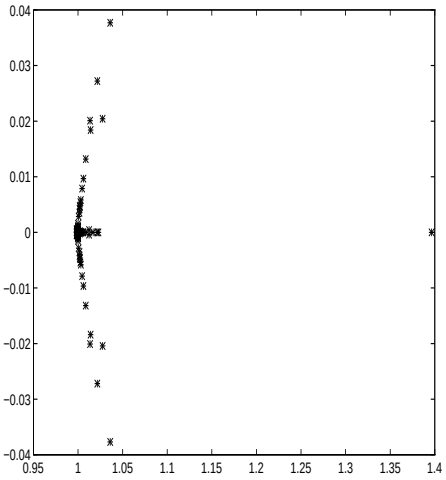

FIG. 11. *Spectra of the Jacobian right-preconditioned by the exact version of the two-stage block discrete projection operator.*

- Continuity or boundeness:

$$\left\| JM^{-1}x \right\| \leq (1+\sigma)\left\|x\right\|, \quad \forall x \in R^n.$$

Under this circunstances, the two above properties predicts the following error bound (convergence factor) for GMRES (a similar characterization has not been proposed for BiCGSTAB)

$$1 - \frac{(1-\sigma)^2}{(1+\sigma)^2} = \frac{4\sigma}{(1+\sigma)^2}.$$

The better the preconditioner the smaller we could expect σ to be. In this sense, it is important to see how the preconditioners developed here are related and predict the convergence of the iterative method. For the sake of simplicity, we assume that we are able to solve exactly (in terms of machine precision) every inner reduced system.

There is evidently a relation between the additive and block Jacobi two-stage preconditioner and between the multiplicative and Gauss-Seidel two-stage preconditioner. By looking at first step of Algorithm 5.2 we can see that the solution of the system is equivalent to the want that we have been solved in terms of J_{pp}^D. In fact,

$$\begin{aligned} \left(R_p^t \widetilde{J}^D R_p\right) p = R_p^t \widetilde{D}^{-1} r &\Leftrightarrow \left(R_p^t P D^{-1} J P^t R_p\right) p = R_p^t P D^{-1} P^t \\ &\Leftrightarrow J_{pp}^D p = r_n^D \end{aligned} \tag{22}$$

Similarly, we can obtain the same correspondence for the saturations. Once a solution for both type of variables is computed, the alternate two-stage preconditioners proceed to improve the residuals by a correction with the preconditioner $\hat{M}$.

It can be shown that

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

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

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

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

In view of (22) we can conclude that the two-stage block Jacobi and the two-stage Gauss-Seidel preconditioners can be expressed as

$$E_{\text{BJ}} = I - JM_{\text{BJ}}^{-1} = I - \widetilde{J}^D\,(t_p + t_s),$$

$$E_{\text{GS}} = I - JM_{\text{GS}}^{-1} = I - \widetilde{J}^D\left(t_p + t_s - t_s\widetilde{J}^D t_p\right).$$

Note that even if we drop the correction step from the alternate two-stage preconditioners, there is not way to reproduce the same action between the two types of preconditioning: the alternate deals with the uncoupled system, whereas the consecutive already deals with the decoupled system that it is expected to be easier to solve.

Hence, we can separate the error propagation associated to $\hat{M}$ from the error propation associated to the whole two-stage preconditioner. To ensure convergence, it is necessary that the norms of each of this errors are bounded above by 1. This imposes the same restrictions to each of the factors involved in the complete error progation expressions. 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 rates of convergence.

It is at this point, that we find a serious limitation, not say drawback, with the use of an extra preconditioner to correct residuals. This situation seems to be more stringent 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 for introducing the operator $\hat{M}$. The computation of new residuals involves one extra matrix vector multiplication and an AXPY operation. This can certainly be computational demanding for large scale problems and for iterative solvers that perform more than one call to the preconditioner for iteration (e.g., BiCGSTAB, CGS).
- In favour to decoupling operators, we have seen that they are effective in clustering the eigenvalues of the original highly indefinite Jacobian matrix. The combinative, additive and multiplicative misuse this property and implicitly reimposes this task to the operator $\hat{M}$. For instance, we require

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

  for the two-stage additive error propagation operator. If $t_p + t_s$ performs a good job preconditiong $\widetilde{J}$, we should expect that $\hat{M}$ does a better or at least, a similar effect. The overall effect is like starting a new preconditioner from scratch that has to eliminate those error frequencies already removed by approximate solution to pressures and saturations. Certainly, this problem can be hard to calibrate for the sake of robust good two-stage preconditioner. 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 significant cost. Although the use of the operator $\hat{M}$ seems to be better justified in the original combinative method, yet it has to capture part of the hyperbolic behavior contained in saturations upon a difficult couple linear problem (recall spectrum pictures in Fig. 2-4,) instead of taking advantage of a reduced saturation problem that can be easily obtained by a better decoupling strategy.

More precisely, a simple computation lead us to

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

Consequently, by taking norms we obtain

$$\|E_{\text{add}}\| \leq \gamma\,(\sigma + \eta)\,,$$

where

$$\begin{aligned} \gamma &= \left\|I - \widetilde{J}\hat{M}^{-1}\right\|; \\ \eta &= \left\|\left(\widetilde{J} - \widetilde{J}^D\right)(t_p + t_s)\right\|; \\ \sigma &= \|E_{\text{BJ}}\| = \left\|I - \widetilde{J}^D M_{\text{BJ}}^{-1}\right\|. \end{aligned}$$

The use of the preconditioning stage suggested by $\hat{M}$ can be only justified in special cases. For example, it can be an operator 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 saturation residuals. In general, it should be designed under simple terms on sequential implementations and with more relaxed bounds if it is suitable 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.

5.6. Efficient implementation. In this section we propose several strategies to enhance the computational efficiency of the two-stage preconditioners. All of the suggestions included here have undergone preliminary evaluation and their implementation in the iterative solvers' code is underway.

In order to decrease the computational requirements of our preconditioners we can use the old but still effective method of line correction. This concept was first introduced in reservoir engineering by Watts [53]. The basic idea is to add the residuals in a given direction (collapsing) and then solve the reduced problem in a lower dimension.

The solution should force the sum of the residuals in the collapsed direction to be zero. Those solutions are then extended (projected back) onto the original dimension and new residuals are formed. Frequently, in order to capture heterogeneities along the collapsed direction, a general relaxation is performed on the new residuals. The collapsing is done, in most cases, along the vertical direction (i.e., depth.) Let us

denote the depth coordinate as k containing nz gridblocks along this direction and, i and j as the plane coordinates with nx and ny gridblocks, respectively. Suppose a lexicographic numbering of nb unknowns with each one of them ocuppying the position (i,j,k). Hence, the collapsing operation can be represented by a rectangular matrix $C \in \mathbb{R}^{nb \times nx*ny}$ such that

$$(C)_{lk} = \begin{cases} 1 & \text{if} \quad l = (k-1)nx * ny + 1, \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, 2, \ldots, nz$. The line correction can be described by the following steps for solving $Ty = r$:

ALGORITHM 5.6.

1. Collapse residuals by means of the operator C^t and solve $(C^t T C)\,\widetilde{w} = C^t r$,
2. Expand solution to the original dimension: $w = C\widetilde{w}$,
3. Compute new residuals $\widetilde{r} = r - Tw$, and,
4. Perform some relaxation steps by a suitable stationary iterative method along the collapsed direction (e.g. line Jacobi, line SOR.) Obtain z.
5. Set $y = w + z$.

We proceed to describe how to incorporate this method in the framework of two-stage preconditioners.

Consider the stage for solving the pressure part in any of the preconditioners defined for the alternate ordering of unknowns. The application of line correction, for instance, in the combinative approach implies the following computation in steps (1-3) in Algorithm 5.1,

$$\widetilde{r} = R_p \widetilde{W}_p r - \widetilde{J} C \left(\widetilde{R}_p^t \widetilde{J}^W \widetilde{R}_p \right)^{-1} \widetilde{R}_p^t \widetilde{W}_p r,$$

and apply some suitable relaxation steps on $\widetilde{r}$. Here, $\widetilde{R}_p = C R_p$.

Note that neither the operator $\widetilde{R}_p$ nor R_p nor C have to be explicitly formed for implementation purposes. Moreover, approximation of the reduced matrices in pressure or saturation can be stored in factored form prior to the execution of the outer linear solver. Application of R_p, C or any other analogous reduction operator can be implicit performed onto the residuals to be preconditioned.

The translation of this procedure to the preconditioners for consecutive ordering of unknowns is straightforward. Note that according to Theorem 4.1, the convergence of any point-, line- or block-type stationary iterative method is guarantee for the resulting decoupled blocks via D^{-1}. This result comes from the fact that from any M-matrix we can produce a convergent weak regular splitting (see [3] for further details.).

Further efficiency enhancements may be realized in the use of inexact methods applied to individual steps of the two-stage preconditioners. Essentially, all alternative and consecutive preconditioners proposed in this work construct the preconditioned residual by (iteratively) solving systems whose coefficient matrices are diagonal blocks of the decoupled Jacobian. We can certainly solve these systems in parallel by an iterative Krylov-subspace method with a block-type preconditioner (e.g., block Jacobi). A variety of domain decomposition approaches, both overlapping and nonoverlapping, can be used to precondition these problems.

In particular, overlapping domain decomposition algorithms, e.g., additive Schwarz, do not require the coarse-grid component in the preconditioned residual computation

TABLE 1

Results for GMRES preconditioned by the nine schemes tested in this work. N_{it}: number of outer iterations; T_s: elapsed time in seconds for the solver iteration; T_p: elapsed time in seconds to form the preconditioner; $N_{i,a}$: average number of inner iterations per unit outer iteration. Preconditioners shown are from top to bottom: Tridiagonal (Tridiag.), Incomplete LU factorization with no infill (ILU(0)), Block Jacobi (BJ), Two-stage Combinative (2-S Comb.), Two-stage Additive (2-S Add.), Two-stage Multiplicative (2-S Mult.), Two-stage Block Jacobi (2-S BJ), Two-stage Gauss-Seidel (2-S GS) and Two-stage Discrete Projection (2-S DP)

Timestep 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	2-S Comb.	390	227.50	0.63	42	>1000	-	-	
×	2-S Add.	42	83.38	1.00	121	30	79.00	0.88	200
8	2-S Mult.	19	38.38	1.00	120	18	47.63	0.88	195
×	2-S BJ	32	19.54	0.02	123	21	24.41	0.02	198
8	2-S GS	16	9.53	0.02	119	13	15.42	0.02	192
	2-S DP	15	15.80	0.03	122	11	20.39	0.04	188

Timestep 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	2-S Comb.	555	527.25	9.63	15	>1000	-	-	
×	2-S Add.	237	680.00	13.50	52	384	1678.13	13.50	80
16	2-S Mult.	103	305.63	13.50	52	90	392.75	13.50	79
×	2-S BJ	52	58.66	0.09	52	37	61.77	0.09	80
16	2-S GS	25	28.77	0.09	52	19	31.68	0.09	78
	2-S DP	21	49.91	0.13	63	17	52.27	0.13	85

for systems involving parabolic convection-difusion with moderate convective component (see [20] for detailed theory on the subject). Results like this are applicable in the algebraic setting of nonsymmetric M-matrices which are diagonally dominant.

Additionally, convergence properties of overlapping schemes are better in 2-D than in 3-D, making them very appealing to solve the 2-D problems arising from the line-correction method. Very robust and highly parallel preconditioners can be formulated this way. Other nonoverlapping domain decomposition methods can also employed but their success for nonsymmetric indefinite systems is more limited than that for the overlapping schemes.

6. Computational experiments. In this section we discuss the results of the numerical experiments shown in Tables 1 and 2, which were designed to test the ideas covered in this work.

The matrices and right hand side vectors for our test problems were generated by the two phase black oil simulator RParSim. Our test model consists of one production and one injection vertical wells located at opposite corners of the reservoir. The permeability is uniform in the areal sense and 15 times higher than that in the vertical

TABLE 2

Results for BiCGSTAB preconditioned by the nine schemes tested in this work. N_{it}: number of outer iterations; T_s: elapsed time in seconds for the solver iteration; T_p: elapsed time in seconds to form the preconditioner; $N_{i,a}$: average number of inner iterations per unit outer iteration. Preconditioners shown are from top to bottom: Tridiagonal (Tridiag.), Incomplete LU factorization with no infill (ILU(0)), Block Jacobi (BJ), Two-stage Combinative (2-S Comb.), Two-stage Additive (2-S Add.), Two-stage Multiplicative (2-S Mult.), Two-stage Block Jacobi (2-S BJ), Two-stage Gauss-Seidel (2-S GS) and Two-stage Discrete Projection (2-S DP).

Timestep 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	2-S Comb.	106	113.75	0.50	40	125	253.50	0.50	81
×	2-S Add.	24	88.75	1.00	118	34	158.38	0.75	183
8	2-S Mult.	13	61.38	1.00	115	14	67.38	0.75	188
×	2-S BJ	23	24.97	0.02	116	24	46.91	0.02	173
8	2-S GS	11	11.91	0.02	115	12	24.15	0.02	177
	2-S DP	10	20.04	0.03	118	14	44.01	0.03	179

Timestep 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	2-S Comb.	170	292.50	9.75	14	180	690.00	12.00	25
×	2-S Add.	68	361.75	13.38	50	61	490.63	13.18	77
16	2-S Mult.	44	238.88	13.38	49	32	255.25	13.38	75
×	2-S BJ	41	83.21	0.09	49	23	68.91	0.09	76
4	2-S GS	17	35.82	0.09	50	11	33.81	0.09	76
	2-S DP	13	56.72	0.12	61	10	58.62	0.12	82

direction. 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 3 time steps and after 2 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 and 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 memory.

The tests included runs made with both GMRES and BiCGSTAB preconditioned with each of the schemes analyzed in this work and, additionally, with three preconditioner of common use in reservoir simulation (particularly the last two), i.e., tridiagonal, ILU(0) (i.e., incomplete LU factorization with no infill) and block Jacobi.

Table 1 shows the results for all the preconditioners applied to GMRES and Table 2 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.

All of the linear systems were solved iteratively until a norm reduction of 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 with tridiagonal preconditiitoning was used as the inner solver in every case, wherever applicable, with a restart of 30 and a linear tolerance equal to that of the outer solves.

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 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 block 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 inspite 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 13.

Comparison between the results for $\Delta t = 0.1$ and those for $\Delta t = 1.0$ show 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 2-S Comb.) 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 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 $N_{i,a}$ show the accumulated average of both the pressure and saturation components whereas the 2-S Combinative 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.

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 the inidividual decoupled pressure and saturation 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 $\hat{M}$ is results in wasted time by the iterative solver.

It should be mentioned that $\hat{M}$ was chosen as an incomplete 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 were always retained (notice that all cases have 4 gridblocks in the z-direction, which is the most rapidly increasing in the numbering scheme of the gridblocks). In spite of the assumed robustness of this global preconditioner, its main effect seems to be the posting of greater elapsed times.

Figure 12 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 show 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 13 shows the same exact arrangement for BiCGSTAB. We note tha, 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 12, lower-right plot, that the consecutive preconditioner produces a dramatically faster convergence than does the fastest of the alterante preconditioners.

7. Conclusions. In this section, we summarize the work we have developed so far and propose the next activities in pursuing the objectives of this dissertation. We have already accomplished the following aspects,

- Description of four novel preconditioners for the solution of coupled equations. Preliminary computational experiments show encouraging results about the effectiveness of these preconditioners.
- We have incorporated in our framework methods, such as the overlapping additive Schwarz and line correction, to speed up implementations of these preconditioners. Meanwhile, we have taken care of their robustness by solving

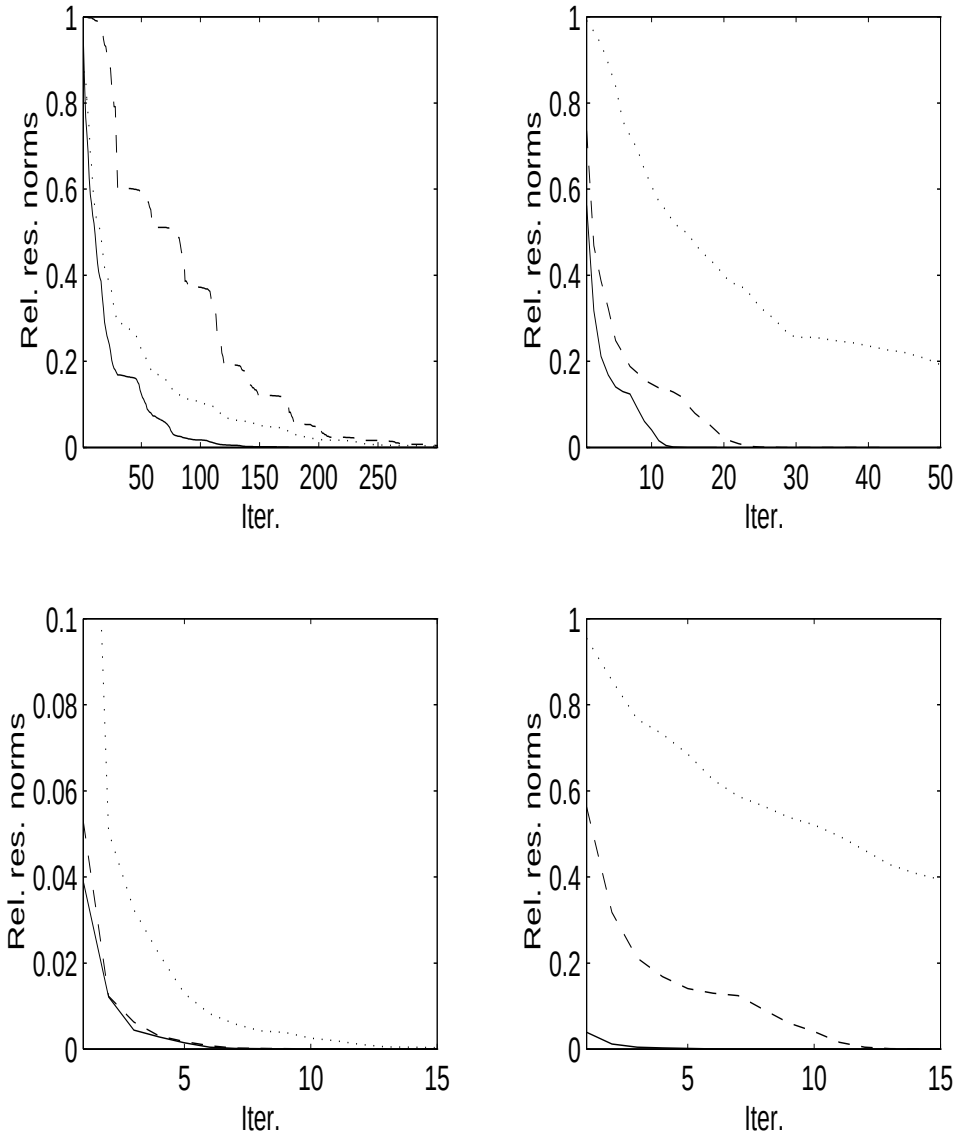

FIG. 12. *Relative residual norms vs. iteration of GMRES for different preconditioners. The performance with different preconditioners are organized in matrix form. Subplot* $(1,1)$*: ILU (dot), Trid(dash), block Jacobi (solid). Subplot* $(1,2)$*: Two-stage combinative (dot), two-stage additive (dash),two-stage multiplicative (solid). Subplot* $(2,1)$*: Two-stage block Jacobi (dot), two-stage Gauss-Seidel (dash), two-stage discrete projection (solid). Subplot* $(2,2)$*: Block Jacobi (dot), two-stage multiplicative (dash), two-stage discrete projection (solid). Problem Size:* $8 \times 8 \times 4$*.* $\Delta t = 0.1$*.*

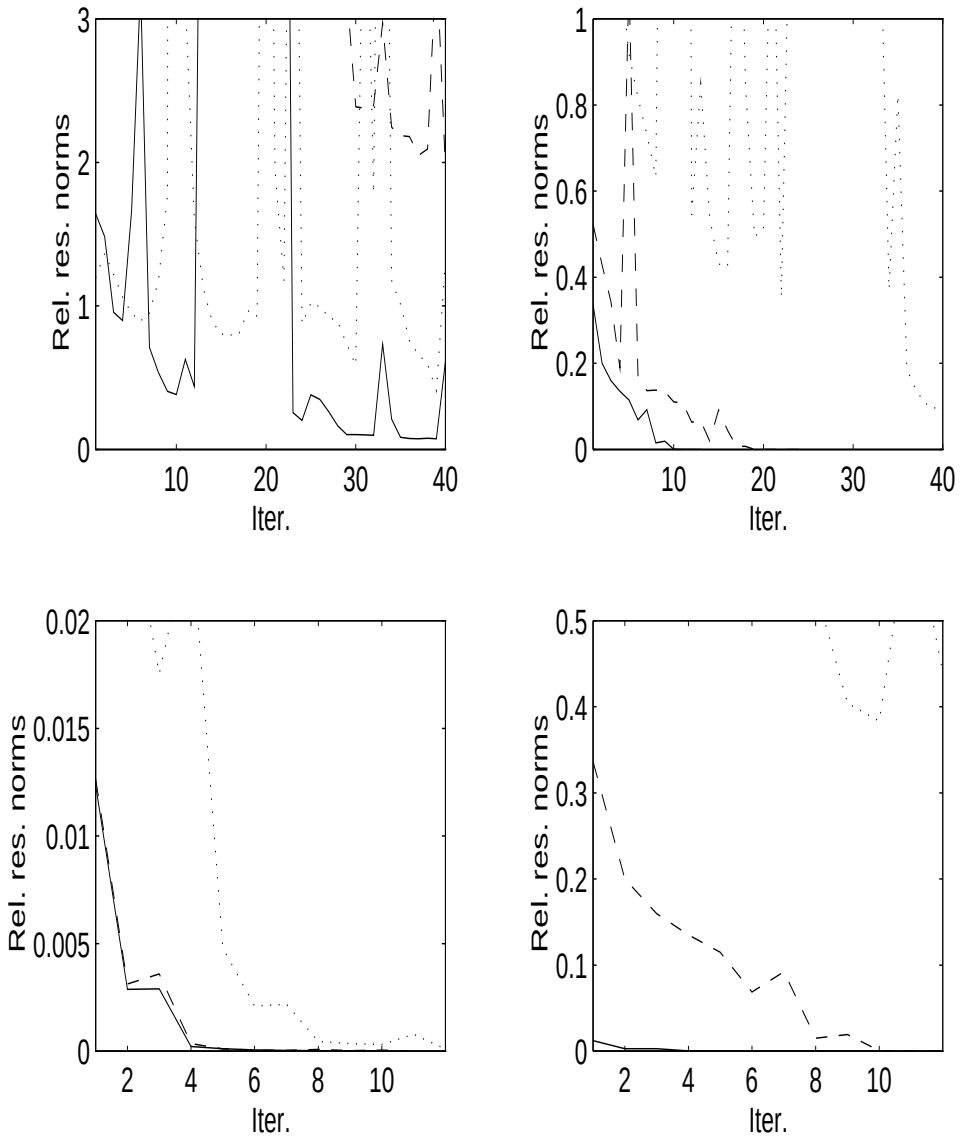

FIG. 13. *Relative residual norms vs. iteration of BiCGSTAB for different preconditioners. The performance with different preconditioners are organized in matrix form. Subplot* $(1,1)$*: ILU (dot), Trid(dash), block Jacobi (solid). Subplot* $(1,2)$*: Two-stage combinative (dot), two-stage additive (dash),two-stage multiplicative (solid). Subplot* $(2,1)$*: Two-stage block Jacobi (dot), two-stage Gauss-Seidel (dash), two-stage discrete projection (solid). Subplot* $(2,2)$*: Block Jacobi (dot), two-stage multiplicative (dash), two-stage discrete projection (solid). Problem Size:* $8 \times 8 \times 4$*.* $\Delta t = 0.1$*.*

decoupled systems by covering all degrees of freedom present in the original coupled system.

- We have also studied secant preconditioners and found encouraging results towards this direction. We have already indicated different avenues that may enhance this approach and as result, save a significant amount of computation for large scale problems.

We are currently investigating the following points,

- Theoretical framework supporting the convergence and performance of our preconditioners.
- Computer evaluation of results under more stringent physical situations and at a larger scale. Here we include the development of parallel implementation of our preconditioners to measure its scalability on a multiprocessor system.

Acknowledgments. The author thank...

REFERENCES

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

[2] M. Allen, G. Behie, and J. Trangenstein, *Multiphase flow in porous media*, in Lectures Notes in Engineering, Springer Verlag, Berlin, 1988.

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

[4] O. Axelsson and P. Vassilevski, *A black box generalized conjugate gradient solver with inner iterations and variable–step preconditioning*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 625–644.

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

[6] R. Bank, T. Chan, W. Coughran, and K. Smith, *The alternate–block–factorization procedure for systems of partial differential equations*, BIT, 29 (1989), pp. 938–954.

[7] 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.

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

[9] G. Behie and P. Vinsome, *Block iterative methods for fully implicit reservoir simulation*, Soc. of Pet. Eng. J., (1982), pp. 658–668.

[10] A. Berman and R. Plemmons, *Nonnegative matrices in the mathematical sciences*, in Classics in Applied Mathematics, SIAM, Philadelphia, 1994.

[11] 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 (1992), pp. 127–162.

[12] P. Bjorstad, W. C. Jr., and E. Grosse, *Parallel domain decomposition applied to coupled transport equations*, in Seventh International Conference on Domain Decomposition Methods for Scientific Computing, D. Keyes and J. Xu, eds., Como, Italy, 1993, American Mathematical Society.

[13] P. Bjørstad and T. Kårstad, *Domain decomposition, parallel computinng and petroleum engineering*, in Domain–based parallelism and problem decomposition methods in computational science and engineering, D. Keyes, Y. Saad, and D. Truhlar, eds., SIAM, 1995, pp. 39–56.

[14] J. Bramble, J. Pasciak, and A. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, in Copper Mountain Conference on Multgrid Methods, 1995.

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

[16] P. N. Brown, A. Hindmarsh, and L. Petzold, *Using Krylov methods in the solution of large–scale differential–algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[17] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Netwon–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

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

[19] X.-C. Cai, W. Gropp, D. Keyes, and M. Tidriri, *Newton–Krylov–Schwarz methods in CFD*, in International Workshop on the Navier–Stokes Equations, R. Rannacher, ed., Braunschweig, 1994, Notes in Numerical Fluid Mechanics, Vieweg Verlag.

[20] T. F. Chan and T. Mathew, *Domain decomposition algorithms*, in Acta Numerica, Cambridge University Press, New York, 1994, pp. 61–143.

[21] C. Dawson, H. Klíe, C. S. Soucie, and M. Wheeler, *The numerical solution of two–phase flow problem in parallel*. In preparation, 1995.

[22] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[23] J. Dendy, *Multigrid methods for three dimensional petroleum reservoir simulation*, in Tenth SPE Symposium on Reservoir Simulation, SPE paper no. 18409, Houston, Texas, 1989.

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

[25] R. Edwing, *The mathematics of reservoir simulation*, in Frontiers in Applied Mathematics, SIAM, Philadelphia, 1983.

[26] S. Eisenstat and H. Walker, *Choosing the forcing terms in an inexact Newton method*, Tech. Rep. TR94–25, Dept. of Computational and Applied Mathematics, Rice University, 1994.

[27] ———, *Globally convergent inexact Newton methods*, SIAM J. Optimization, 4 (1994), pp. 393–422.

[28] H. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

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

[30] P. Forsyth and P. Sammon, *Practical considerations for adaptive implicit methods in reservoir simulation*, J. of Comp. Physics, 62 (1986), pp. 265–281.

[31] R. Freund, G. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, in Acta Numerica, Cambridge University Press, New York, 1991, pp. 57–100.

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

[33] S. Gomez and J. Morales, *Performance of Chebyshev iterative method, GMRES and ORTHOMIN on a set of oil reservoir simulation problems*, in Mathematics for Large Scale Computing, In J.C. Diaz, New York, Basel, 1989, pp. 265–295.

[34] R. Hanby, D. Sylvester, and J. Chew, *A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow*, Tech. Rep. TR94–246, University of Manchester, 1994.

[35] V. Haroutunian, M. 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 (1993), pp. 323–348.

[36] J. Killough and M. 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.

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

[38] H. Klíe, M. Ramé, and M. Wheeler, *Krylov–secant methods for solving systems of nonlinear equations*, Tech. Rep. TR95–27, Dept. of Computational and Applied Mathematics, Rice University, 1995.

[39] C. Mattax and R. Dalton, *Reservoir Simulation*, vol. 13, SPE–Monograph Series, Richardson, TX, 1990.

[40] R. Nabben, *A new application for generalized M–matrices*, in Numerical Linear Algebra, proceedings of the conference in Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan, and R. Varga, eds., Walter de Gruyter, 1993.

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

[42] J. Nolen and D. Berry, *Test on the stability and time-step sensitivity of semi-implicit reservoir simulation techniques*, Trans. SPE of AIME, 253 (1973), pp. 253–266.

[43] Y. Saad, *A flexible inner–outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[44] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[45] I. Taggart and W. Pinczewski, *The use of higher-order differencing techniques in reservoir simulation*, SPE Reservoir Engineering, (August 1987), pp. 360–372.

[46] R. Teigland and G. Fladmark, *Cell centered multigrid methods in porous media flow*, in Multigrid methods III : proceedings of the 3rd European Multigrid Conference, Birkhauser Verlag, 1991.

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

[48] H. van der Vorst, *BICGSTAB: a fast and smoothly convergent variant of BI–CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.

[49] H. van der Vorst and C. Vuik, *GMRESR: A Family of Nested GMRES Methods*, Tech. Rep. TR91–80, Technological University of Delft, 1991.

[50] C. Vuik, *Further experiences with GMRESR*, Tech. Rep. TR92–12, Technological University of Delft, 1992.

[51] J. Wallis, *Constrained residual acceleration of conjugate residual methods*, in Eigth SPE Symposium on Reservoir Simulation, SPE paper no. 13536, Dallas, Texas, 1985.

[52] ———, *Two-step preconditioning.* Private Communication, 1993.

[53] J. Watts, *A method of improving line succesive overrelaxation in anisotropic problems–a theoretical analysis*, Soc. of Pet. Eng. J., (1973), pp. 105–118.

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

[55] J. Wheeler and R. 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.