Advanced Solver Methods for Subsurface Environmental Problems *

Héctor Klíe, and Mary F. Wheeler

The Center for Subsurface Modeling (CSM)
The Institute for Computational Engineering and Sciences (ICES)
The University of Texas at Austin, Austin, TX, 78712

Contents

*Keywords and phrases: residual corrections, preconditioning, decoupling, coupled systems, two-stage preconditioning, Newton-Krylov, Krylov-secant, two phase flow, groundwater flow, Richards' equation;
Author's e-mail: {klie,mfw}@ices.utexas.edu; UT, Austin.

Abstract

The present report outlines recent results relative to the development of new linear and nonlinear approaches for solving groundwater flow problems. Efficient and robust simulation of complex processes in environmental applications is crucial for understanding, forecasting and performing opportune and reliable decisions. This research effort focuses in developing efficient solvers for two main models in groundwater flow: Richards' equation and air-water coupled equation systems. To achieve this goal, two main streams of research are presented:(a) High-order Newton-Krylov methods, and (b) Decoupling and constraint operators for preconditioning coupled systems of nonlinear equations. Numerical results reveal that these approaches have high potentials for dealing with stringent physical situations.

1 Introduction

High-order Krylov-Newton methods refer to a novel idea originally presented in [12] for solving oil reservoir applications and that has the potential to accelerate the nonlinear iterations by exploiting the Krylov information generated by the GMRES iterative method. This idea is for first time applied to solve Richards' equation for modelling unsaturated flow.

Decoupling and constraint operators are the basis for developing efficient two-stage preconditioners and super-coarsening multigrid methods [5, 12, 16, 17]. The framework can be used as a starting point for generating more sophisticated preconditioning approaches when the discretized model is coupled and involves the solution of variables with different physical behavior such as pressure head and saturation in a water-oil system.

This report is organized as follows. We follow with a brief presentation of the Newton-Krylov framework that encompasses the solution of Richards' equation and the two-phase problems in porous media. In sections 3 and 4, we describe how this framework can be extended to increase nonlinear convergence by means of secant updates restricted to the Krylov subspace. Sections 5, 6, 7 and 8 focus the discussion on decoupling and constraint operators and how they can be used as a basis for two-stage preconditioning and semi- supercoarsening multigrid. Section 9 show some preliminary numerical results. We conclude with some extra discussions and possible extensions to this work. Extended descriptions of these results will be available in two separate ICES technical reports [13, 14].

2 Newton-Krylov framework

The solution of the nonlinear system of equations

$$F(u) = 0, \tag{1}$$

where $F : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}^n$, is cornerstone in many scientific and engineering applications. A variety of discretization techniques applied to systems of coupled nonlinear partial differential equations gives rise to an algebraic system such as (1), where u is the unknown vector representing the nodal values of the state variable.

In our particular case, this nonlinear system of equations arises from discretizing either Richards' equation or the coupled partial differential equations describing the air-water system. The nonlinearities in both equations are present in the temporal term and the diffusive term and when fully implicit schemes are employed the transient and steady-state cases lead to a nonlinear algebraic system such as (1). Newton's method implies the solution of the Jacobian system

$$J(u)\,\delta u = -F(u), \tag{2}$$

where $F(u)$ and $J(u)$ denote the evaluation of the function at $u \in \mathbb{R}^n$ and its derivative at any Newton step, respectively.

Due to the high cost or absence of derivatives for the explicit construction of the Jacobian J, the action of this operator onto vector (in the form of matrix-vector products) can be performed by finite differences. This is, in fact, one of the most appealing features of Newton-Krylov methods. These methods are also amenable to be globalized by line-search or trust region strategies and for dynamic control of linear tolerances (i.e., *forcing terms*). The reader is referred to [15] to obtain an state of the art discussion on the subject. Specific theory supporting the description of the line-search backtracking method and the selection of the forcing terms to control the convergence of the linear solver at each nonlinear step can be found in [9, 11, 21]. A globalized Newton-Krylov method with a line-search backtracking procedure can be stated as follows:

Algorithm 2.1 (Newton-Krylov algorithm with forcing terms and line-search backtracking)

1. Let $u^{(0)}$ be an initial guess
2. For $k = 0, 1, 2, \ldots$ until convergence do

2.1 Choose $\eta^{(k)} \in [0,1)$, $t \in (0,1)$

2.2 Using some Krylov subspace iterative method, compute a vector $s^{(k)}$ satisfying

$$J^{(k)}s^{(k)} = -F^{(k)} + r^{(k)}, \tag{3}$$

with $\frac{\|r^{(k)}\|}{\|F(u^{(k)})\|} \leq \eta^{(k)}$.

2.3 While $\left\|F\left(u^{(k)} + s^{(k)}\right)\right\|_2 > \left[1 - t\left(1 - \eta^{(k)}\right)\right]\left\|F^{(k)}\right\|$ do

2.3.1 Choose λ to minimize a quadratic polynomial over $\left[\underline{\lambda}, \overline{\lambda}\right] \subset (0,1)$ that interpolates

$$F(\lambda) = \left\|F\left(u^{(k)} + \lambda s^{(k)}\right)\right\|^2.$$

2.3.2 Update $s^{(k)} = \lambda s^{(k)}$.

2.4 Update $\eta^{(k)} = 1 - \lambda\left(1 - \eta^{(k)}\right)$.

3. Update solution $u^{(k+1)} = u^{(k)} + s^{(k)}$.

Some observations are in order.

- The parameter t is usually small, $t = 10^{-4}$ (i.e., a small margin between the predicted and actual reduction of $\left\|F^{(k)}\right\|$ is assumed small).
- An interval value $\left[\underline{\lambda}, \overline{\lambda}\right] = [.1, .5]$ is typically chosen.
- A quadratic interpolant p, is constructed in such a way that $p(0) = F(0) = \left\|F^{(k)}\right\|^2$, $p(1) = F(1) = \left\|F^{(k)} + s^{(k)}\right\|^2$ and $p'(0) = F'(0) = 2\left\langle F^{(k)}, J^{(k)}s^{(k)}\right\rangle$.

The forcing term $\eta^{(k)}$ is selected according to the following criterion:

$$\eta^{(k)} = \min\left\{\eta_{\max}, \max\left\{\widetilde{\eta}^{(k)}, \left(\eta^{(k-1)}\right)^2\right\}\right\},$$

where

$$\widetilde{\eta}^{(k)} = \frac{\left|\left\|F^{(k)}\right\| - \left\|F^{(k-1)} + J^{(k-1)}s^{(k-1)}\right\|\right|}{\left\|F^{(k-1)}\right\|}.$$

The choice of $\widetilde{\eta}^{(k)}$ reflects the agreement between and its linear model at the previous iteration. Thus, the linear solver tolerance is larger when

the Newton step is less likely to be productive and smaller when the step is more likely to lead to a good approximation. This operation is performed once the backtracking procedure has returned the adequate nonlinear step size and update of the solution.

3 The family of rank–one solvers

Rank-one updates for solving nonlinear equation are sometimes referred as secant or *quasi-Newton methods* since no "true" Newton equation is ever formed throughout all iterations. These updates obey a secant condition that has to be satisfied by the new Jacobian approximation. The best of these methods still seems to be the first one originally introduced by Broyden (see e.g., [6]).

3.1 Broyden's method

Given $u \approx u^*$ and $A \approx J(u)$, we can find an approximate new Newton step, u^+, by

$$u^+ = u - A^{-1}F(u). \tag{4}$$

Broyden's method computes a new A^+ by means of the following rank-one update

$$A^+ = A + \frac{[F^+(u) - F(u) - As]\, g^t}{g^t s}, \tag{5}$$

which, whenever the approximated Jacobian equation is solved exactly, i.e. $As = -F(u)$, it reduces to

$$A^+ = A + \frac{F^+(u)\, g^t}{g^t s},$$

for $g^t s \neq 0$. The vector g can be chosen in several ways. For instance, when $g \equiv s$, we obtain the "good Broyden's update" and when $g \equiv A^t [F^+(u) - F(u)]$, we have the "bad Broyden's update". The algorithm can be depicted as follows:

Algorithm 3.1 (Nonlinear Broyden)

1. Give an initial guess $u^{(0)}$ and Jacobian approximation A_0.
2. For $k = 0, 1, \ldots$ until convergence do
   2.1 Solve $A^{(k)} s^{(k)} = -F^{(k)}$.

2.2 Update solution $u^{(k+1)} = u^{(k)} + s^{(k)}$.

2.3 $q^{(k)} = F^{(k+1)} - F^{(k)}$.

2.4 $A^{(k+1)} = A^{(k)} + \frac{\left(q^{(k)} - A^{(k)}s^{(k)}\right)\left(s^{(k)}\right)^t}{\left(s^{(k)}\right)^t s^{(k)}}$.

3.2 The Nonlinear Eirola-Nevanlinna (NEN) method

The nonlinear Eirola-Nevanlinna (NEN) was proposed by [25] as the nonlinear counterpart of the linear EN algorithm [8]. This method is a generalization of Broyden's method for solving linear/nonlinear equations. Interestingly enough, the algorithm has connections with the GMRES algorithm [23]. The NEN algorithm can be regarded as the composition of two Broyden's iterations as the following presentation suggests:

Algorithm 3.2 (Nonlinear EN)

1. Give an initial guess $u^{(0)}$ and Jacobian approximation M_0.
2. For $k = 0, 1, \ldots$ until convergence do

   2.1 Solve $M^{(k)}s^{(k)} = -F^{(k)}$.

   2.2 $q^{(k)} = F^{(k+1)} - F^{(k)}$.

   2.3 $M^{(k+1)} = M^{(k)} + \frac{\left(q^{(k)} - M^{(k)}s^{(k)}\right)\left(s^{(k)}\right)^t}{\left(s^{(k)}\right)^t s^{(k)}}$.

   2.4 Solve $M^{(k+1)}\widetilde{s}^{(k)} = -F^{(k)}$.

   2.5 Update solution $u^{(k+1)} = u^{(k)} + \widetilde{s}^{(k)}$.

Notice that the direction computed by the NEN algorithm is a linear combination of the direction delivered by Broyden's method and an extra direction coming from step 2.4. It can be shown that the NEN algorithm converges n-step q-quadratically for n-dimensional problems. Therefore, the NEN method converges twice faster than Broyden's method.

4 High-order Newton-Krylov methods

Consider again A as an approximation to the current Jacobian matrix J. We are interested in looking at a minimum change to A consistent with $A^+s = F^+ - F$ restricted to the underlying Krylov subspace. A basis for this subspace arises as result of using an iterative linear solver such as GMRES for solving the approximated Jacobian system with A.

4.1 Secant updates onto the Krylov subspace

The Arnoldi factorization provides valuable information that should not be discarded at all every time a GMRES solution starts over. We now show how to reflect secant updates on the Jacobian matrix without altering the current Krylov basis. For the sake of simplicity, let us omit the sources of inexactness induced by the use of GMRES whose relative residuals are supposed to converge at a predefined tolerance (i.e., to a prescribed forcing term value).

Consider the solution to the following approximated Jacobian equation at the kth nonlinear iteration

$$A^{(k)}s^{(k)} = -F^{(k)}, \tag{6}$$

with l steps of the GMRES algorithm. This linear solution can be regarded as embedded in an inexact Broyden's method. Let $s_l^{(k)} = s_0^{(k)} + V^{(k)}y^{(k)}$ be the solution obtained. The associated Krylov subspace for this problem is given by $\mathcal{K}_l^{(k)}\left(A^{(k)}, r_0^{(k)}\right)$. Now, we wish to use the information gathered during the solution of (6) to provide an approximation to the system

$$A^{(k+1)}s^{(k+1)} = -F^{(k+1)}, \tag{7}$$

with corresponding Krylov basis $\mathcal{K}_l^{(k)}(A^{(k+1)}, r_0^{(k+1)})$. Clearly, in general we can not guarantee that $\mathcal{K}_l^{(k+1)}(A^{(k+1)}, r_0^{(k+1)}) = \mathcal{K}_l^{(k)}(A^{(k)}, r_0^{(k)})$. However, rank-one updates onto the corresponding Arnoldi factorization of (6) can be done without destroying the Krylov basis. That is,

$$\left(A^{(k)} + V^{(k)}zw^t\left(V^{(k)}\right)^t\right)V^{(k)} = V^{(k)}\left(H_l^{(k)} + zw^t\right) + h_{m+1,m}^{(k)}v_{m+1}^{(k)}e_l^t, \tag{8}$$

or equivalently,

$$\left(V^{(k)}\right)^t\left[A^{(k)} + V^{(k)}zw^t\left(V^{(k)}\right)^t\right]V^{(k)} = H_l^{(k)} + zw^t, \tag{9}$$

for any vectors $z, w \in \mathbb{R}^l$. Expression (9) suggests a clearer way to update $H_l^{(k)}$ rather than $A^{(k)}$. Note that the current Jacobian approximation appears to be updated by a rank-one matrix whose range lies on $\mathcal{K}_l^{(k)}\left(A^{(k)}, r_0^{(k)}\right)$.

Before proceeding, it would be convenient to express the secant equation

$$A^{(k+1)}s^{(k)} = F^{(k+1)} - F^{(k)}, \tag{10}$$

in terms of a solution lying strictly on the Krylov subspace. Otherwise, this would introduce an implicit secant equation in terms of $A^{(k+1)}$. To remove the shift from the origin, we reformulate (6) as

$$A^{(k)}s^{(k)} = -F^{(k)} - A^{(k)}s_0^{(k)} = r_0^{(k)}, \tag{11}$$

and redefine the final solution as $s_l^{(k)} = V^{(k)}y^{(k)}$, that is, as if the initial guess were zero. Obviously, the associated Krylov basis is the same depicted above. Therefore, the secant equation (10) becomes

$$A^{(k+1)}s^{(k)} = F^{(k+1)} + r_0^{(k)}, \tag{12}$$

for $s^{(k)} = V^{(k)}y^{(k)}$. Multiplying both sides by $\left(V^{(k)}\right)^t$ it readily follows that $H_l^{(k+1)}$ should satisfy the following secant equation

$$H_l^{(k+1)}y^{(k)} = \left(V^{(k)}\right)^t F^{(k+1)} + \beta e_1, \tag{13}$$

where $\beta = \left\|r_0^{(k)}\right\|$. Hence, the Krylov subspace projected version of the secant equation (10) can be written as

$$H_l^{(k+1)} = H_l^{(k)} + \frac{\left(\left(V^{(k)}\right)^t F^{(k+1)} + \beta e_1 - H_l^{(k)}y^{(k)}\right)\left(y^{(k)}\right)^t}{\left(y^{(k)}\right)^t y^{(k)}}. \tag{14}$$

4.2 The nonlinear KEN algorithm

We are now in a position to describe an inexact NEN algorithm that exploits the information left behind by the GMRES method. Hence, we introduce the nonlinear Krylov-Eirola-Nevanlinna (nonlinear KEN) algorithm as follows.

Algorithm 4.1 (Nonlinear KEN)

1. Give an initial guess $u^{(0)}$ and Jacobian approximation $A^{(0)}$.
2. For $k = 0, 1, \ldots$ until convergence do

   2.1 $\left[s^{(k)}, y^{(k)}, H_l^{(k)}, V_l^{(k)}, h_{m+1,m}^{(k)}, \beta \equiv \left\|r_0^{(k)}\right\|\right] = \text{GMRES}\left(A^{(k)}, -F^{(k)}, s^{(k)}\right).$

   2.2 $q^{(k)} = \left(V_l^{(k)}\right)^t F^{(k+1)} + \beta e_1.$

   2.3 $H_l^{(k+1)} = H_l^{(k)} + \frac{\left(q^{(k)} - H_l^{(k)}y^{(k)}\right)\left(y^{(k)}\right)^t}{\left(y^{(k)}\right)^t y^{(k)}}.$

2.4 Solve

$$\min_{y\in\mathcal{K}_l\left(A^{(k)},r_0^{(k)}\right)} \left\|\beta e_1+\overline{H}_l^{(k+1)}y\right\|,$$

$$\text{with } \overline{H}_l^{(k+1)}=\begin{pmatrix} H_l^{(k+1)} \\ h_{m+1,m}^{(k)}e_l^t \end{pmatrix}.$$

Denote its solution by $\widetilde{y}^{(k)}$.

2.5 $\widetilde{s}^{(k)}=V_l^{(k)}\widetilde{y}^{(k)}$.

2.6 Perform

$$A^{(k+1)}=A^{(k)}+P^{(k)}\frac{\left[F^{(k+1)}+r_0^{(k)}-A^{(k)}s^{(k)}\right]\left(s^{(k)}\right)^t}{\left(s^{(k)}\right)^t s^{(k)}},$$

with $P^{(k)}=V^{(k)}\left(V^{(k)}\right)^t$.

2.7 $u^{(k+1)}=u^{(k)}+\widetilde{s}^{(k)}$.

4.3 A high-order Newton–Krylov algorithm

A even faster version of the nonlinear KEN algorithm can be attained by performing rank-one updates of the Hessenberg matrix as far as possible before making the next GMRES call. The extend of these updates is determined by the capability the residual minimization problem in producing descent directions. In this opportunity, we abandon simultaneous Jacobian and Hessenberg updates and check instead if a sufficient amount of decrease of $\|F\|$ is delivered by verifying the condition state at step 2.3 of Algorithm 2.1.

This presentation allows us to illustrate further uses of the Krylov–Broyden update (14)in the context of Newton's method. We stress that further updates to the Hessenberg matrix may result in relative less overhead for a higher order version of the inexact Newton method than a possible faster nonlinear KEN algorithm. The point is that the latter one requires simultaneous updates of $H_l^{(k)}$ and $A^{(k)}$ which may readily increase the total number of updates. Of course, this may be a desirable situation in terms of rapid convergence updates but it may turn out to be expensive in terms of computer memory use. Letting $\beta=\left\|r_0^{(k)}\right\|$, the algorithm can be outlined as follows.

Algorithm 4.2 (High-Order Newton–Krylov, HONK)

1. Give an initial guess $u^{(0)}$ and define r_{max}.
2. For $k = 0, 1, \ldots$ until convergence do

   2.1 $\left[s^{(k)}, y^{(k)}, H_l^{(k)}, V_l^{(k)}, h_{l+1,l}^{(k)}, \beta\right] = \text{GMRES}\left(J^{(k)}, -F^{(k)}, s^{(k)}\right)$.

   2.2 $r = 0$.

   2.3 Repeat

   2.3.1 $q^{(k+r)} = \left(V_l^{(k)}\right)^t F^{(k+r+1)} + \beta e_1$.

   2.3.2 $H_l^{(k+r+1)} = H_l^{(k+l)} + \dfrac{\left(q^{(k+r)} - H_l^{(k+r)} y^{(k+r)}\right)\left(y^{(k+r)}\right)^t}{\left(y^{(k+r)}\right)^t y^{(k+r)}}$.

   2.3.3 Solve

$$\min_{y \in \mathcal{K}_l\left(A^{(k)}, r_0^{(k)}\right)} \left\| \beta e_1 + \overline{H}_l^{(k+1+r)} y \right\|,$$

$$\text{with } \overline{H}_l^{(k+r+1)} = \begin{pmatrix} H_l^{(k+r+1)} \\ h_{l+1,l}^{(k)} e_l^t \end{pmatrix}.$$

   Denote its solution by $y^{(k+r+1)}$.

   2.3.4 $s^{(k+l+1)} = V_l^{(k)} y^{(k+l+1)}$.

   2.3.5 $t = r + 1$.

   2.4 Until ($r = r_{max}$) OR $s^{(k+l)}$ is not a decreasing step for $\|F^{(k+r)}\|$.

   2.5 if $s^{(k+r)}$ is a decreasing step for $\|F^{(k+r)}\|$ then

   2.5.1 $u^{(k+1)} = u^{(k)} + s^{(k+r_{max})}$.

   2.6 else

   2.6.1 $u^{(k+1)} = u^{(k)} + s^{(k+r-1)}$.

3. EndFor

This algorithm can be devised as a variant of the composite Newton's method that seek chord directions belonging to the underlying Krylov subspace. The faster version of the nonlinear KEN algorithm can be easily stated from the above presentation by just including the Krylov-Broyden update of $A^{(k)}$ within the repeat loop statement 2.3. This version should be appealing in situations where Broyden's method or the nonlinear EN are effective compared to Newton's method. Figure 1 summarizes the functionality of the HONK algorithm.

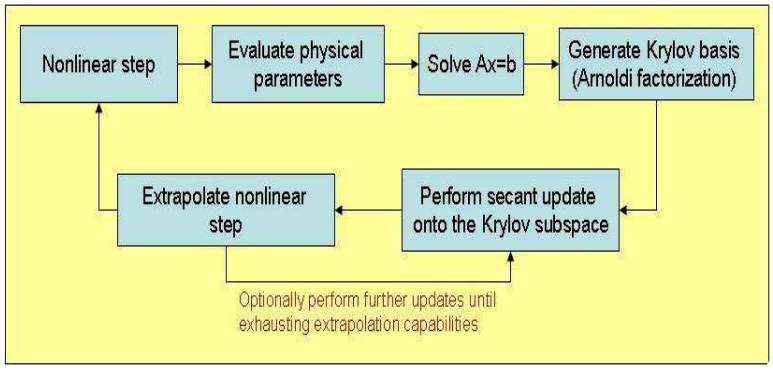

Figure 1: Schematic representation of the HONK method.

5 The air-water case: A coupled algebraic system of equations

By applying mixed finite element (equivalent to cell-centered finite differences under suitable conditions, see e.g. [7]) in space and a Euler formulation in time, we arise to a 2×2 block linear system for solving the changes in water pressures and water saturations. More precisely, equation 2 turns out to be

$$J(u)\,\delta u = \begin{pmatrix} J_{pp} & J_{ps} \\ J_{sp} & J_{ss} \end{pmatrix} \begin{pmatrix} \delta P_w \\ \delta S_w \end{pmatrix} = - \begin{pmatrix} R_w \\ R_g \end{pmatrix} = -F(u)\,, \tag{15}$$

where R_w, R_g represents the value of the water equation residual and gas equation residual, respectively. Here, the nonlinear function $F(u) : \mathbb{R}^m \to \mathbb{R}^m$, where m is twice the number of gridblocks of the discretized domain, holds the value of the residuals at the current iteration level. The matrix $J(u)$ is known as the Jacobian matrix and represents the derivatives of the nonlinear function $F(u)$ with respect each value of u at every gridblock ijk. We now provide general description of the linear systems (i.e., Newton equation) arising in step 2.2 of Algorithm 2.1.

According to (15) each block $J_{ij}, i, j = p, s$ is of size $nb \times nb$, where nb, is the total number of grid blocks, so that $m = 2 * nb$. Each group of unknowns is numbered in a sequential lexicographic fashion: the pressure unknowns are numbered from one through the total number of grid blocks nb and the saturations are numbered from $nb+1$ through m. These observations follow:

- The matrix J is indefinite and nonsymmetric.

- Each block has the same 7-diagonal structure.
- The block J_{pp}, containing pressure coefficients, resembles the structure of a purely elliptic problem.
- The block J_{ps}, has a structure similar to that of a discretized first-order hyperbolic problem in the gas phase saturations
- The block J_{sp} has the coefficients of a convection-free parabolic problem in the non-wetting phase pressure
- The block J_{ss} represents a parabolic (convection-diffusion) problem for water saturations.
- The blocks $J_{pp}J_{ps}, J_{sp}$ and $-J_{ss}$ show diagonal dominance, positive diagonal entries and nonpositive off-diagonal entries along their diagonal (i.e., they are Z-matrices) and are irreducible (see e.g., [1] on further implications of these properties).

6 Decoupling Operators

Consider the Jacobian system shown in (15) and let us define

$$D = \begin{pmatrix} D_{pp} & D_{sn} \\ D_{sp} & D_{ss} \end{pmatrix} = \begin{pmatrix} diag(J_{pp}) & diag(J_{ps}) \\ diag(J_{sp}) & diag(J_{ss}) \end{pmatrix}, \tag{16}$$

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

$$\begin{aligned} D^{-1}J &= \begin{pmatrix} \Delta^{-1} & 0 \\ 0 & \Delta^{-1} \end{pmatrix} \begin{pmatrix} D_{ss}J_{pp} - D_{sn}J_{sp} & D_{ss}J_{ps} - D_{sn}J_{ss} \\ D_{pp}J_{sp} - D_{sp}J_{pp} & D_{pp}J_{ss} - D_{sp}J_{ps} \end{pmatrix} \\ &\equiv \begin{pmatrix} J^D_{pp} & J^D_{sn} \\ J^D_{sp} & J^D_{ss} \end{pmatrix} = J^D, \end{aligned} \tag{17}$$

where $\Delta = D_{pp}D_{ss} - D_{sn}D_{sp}$. We can observe that the main diagonal entries of the main diagonal blocks are equal to one. Conversely, the main diagonal entries of the off-diagonal blocks are 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. In physical terms, the decoupling operator tends to approximate pressure coefficients as if saturations derivatives were neglected in the transmissibility coefficients. Hence, this is like "time-lagging" or evaluating some transmissibilities explicitly.

The above decoupling operator admits an alternate or point-blockwise representation. We can associate the smaller matrix blocks 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 concentration unknown at the same grid block. Let P be the matrix that perform such permutation and define

$$\widetilde{J} = PJP^t = \begin{pmatrix} \widetilde{J}_{11} & \widetilde{J}_{12} & \cdots & \widetilde{J}_{1nb} \\ \widetilde{J}_{21} & \widetilde{J}_{22} & \cdots & \widetilde{J}_{nb} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{J}_{nb1} & \widetilde{J}_{nb2} & \cdots & \widetilde{J}_{nb,nb} \end{pmatrix}, \tag{18}$$

where

$$\widetilde{J}_{ij} = \begin{pmatrix} (J_{pp})_{ij} & (J_{ps})_{ij} \\ (J_{sp})_{ij} & (J_{ss})_{ij} \end{pmatrix}, \tag{19}$$

is the 2×2 matrix representing the coupling between unknowns. It clearly follows for D that

$$\widetilde{D} = PDP^t \Longleftrightarrow \widetilde{D}^{-1} = PD^{-1}P^t. \tag{20}$$

Hence, $\widetilde{D}^{-1}$ is a block diagonal matrix whose blocks are the inverses of the Jacobian blocks associated to the local problem at each grid block. That is,

$$\widetilde{D}^{-1} = \begin{pmatrix} \widetilde{J}_{11}^{-1} & 0 & \cdots & 0 \\ 0 & \widetilde{J}_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \widetilde{J}_{nb,nb}^{-1} \end{pmatrix}. \tag{21}$$

Clearly, $\widetilde{D}^{-1}\widetilde{J} = PD^{-1}JP^t$. In the case of the combinative approach, the coefficients introducing the coupling with pressures are zeroed out within the grid block by Gaussian elimination so that corresponding coefficients at neighboring grid blocks are expected to become small. To be more precise, let

$$\widetilde{W}_p = \begin{pmatrix} (\widetilde{W}_p)_1 & 0 & \cdots & 0 \\ 0 & (\widetilde{W}_p)_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & (\widetilde{W}_p)_{nb} \end{pmatrix}, \tag{22}$$

where

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

with $e_1 = (1,0)^t$. 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

$$(\widetilde{W}_p)_i \widetilde{J}_{ii} = \widetilde{J}_{ii} - e_1 e_1^t \widetilde{J}_{ii} + \left(e_1^t \widetilde{J}_{ii} e_1\right) e_1 e_1^t = \begin{pmatrix} \left(\widetilde{J}_{pp}\right)_{i,i} & 0 \\ \left(\widetilde{J}_{sp}\right)_{i,i} & \left(\widetilde{J}_{ss}\right)_{i,i} \end{pmatrix}. \quad (23)$$

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 [24].

Letting $\widetilde{J}^{W_p} = \widetilde{W}_p \widetilde{J}$, the consecutive counterpart W_p of the alternate representation of the operator $\widetilde{W}_p$ is given by

$$\begin{aligned} J^{W_p} \equiv W_p J = P\widetilde{J}^{W_p} &= \left(P\widetilde{W}_p P^t\right)\left(P\widetilde{J}P^t\right) = \left(P\widetilde{W}_p P^t\right) J \\ &\equiv \begin{pmatrix} J_{pp}^{W_p} & J_{ps}^{W_p} \\ J_{sp}^{W_p} & J_{ss}^{W_p} \end{pmatrix}. \end{aligned} \quad (24)$$

Note that the lower blocks are unmodified as well as the main diagonal of the resulting pressure block, that is, $J_{sp}^{W_p} = J_{sp}$ and $J_{ss}^{W_p} = J_{ss}$.

7 Two-stage preconditioners

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

Therefore, to obtain the pressure system, the following operation can be applied

$$R_p^t J^{W_p} R_p = J_{pp}^{W_p}. \quad (25)$$

7.1 Combinative two-stage preconditioner

Consider the two-stage combinative preconditioner M_{Comb} expressed as the solution of the preconditioned residual equation $M_{Comb}v = r$. Then the action of the preconditioner M_{Comb} is described by the following steps,

Algorithm 7.1 (Two-stage Combinative, 2SComb)

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

$$v = M_{\text{Comb}}^{-1} r = \hat{M}^{-1}\left[I - \left(\widetilde{J} - \hat{M}\right) R_p \left(R_p^t \widetilde{J}^{W_p} R_p\right)^{-1} R_p^t \widetilde{W}_p\right] r. \qquad (26)$$

7.2 Additive two-stage preconditioner

With the use of the operator $\widetilde{D}^{-1}$ and incorporating the solution to a reduced saturation system in addition to the solution to the reduced pressure system we can improve the quality of the previous preconditioner. We propose two different ways to accomplish this: additively and multiplicatively. In the following we present both procedures for computing the preconditioned residual $v = M_{\text{add}}^{-1} r$ and $v = M_{\text{mult}}^{-1} r$. We denote $\widetilde{J}^D = \widetilde{D}^{-1}\widetilde{J}$. The additive two-stage preconditioner (2SAdd) is given by

Algorithm 7.2 (Two-stage Additive, 2SAdd)

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

7.3 Multiplicative two-stage preconditioner

The multiplicative two-stage preconditioner (2SMult) proposes the sequential treatment of the partially preconditioned residuals instead. In algorithmic terms it is given by

Algorithm 7.3 (Two-stage Multiplicative, 2SMult)

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 $\hat{r} = \widetilde{D}^{-1} r - \widetilde{J}^D p$.
4. Solve the reduced concentration system $\left(R_s^t \widetilde{J}^D R_s\right) c = R_s^t \hat{r}$ and denote its solution by $\hat{c}$.
5. Obtain expanded solutions $c = R_s \hat{c}$.
6. Compute new residual $w = \widetilde{D}^{-1} r - \widetilde{J}^D (c + p)$.
7. Precondition and correct $v = \hat{M}^{-1} w + c + p$.

Assuming that both reduced pressures and saturations are solved exactly and introducing the notation

$$t_l \equiv R_l \left(R_l^t \widetilde{J}^D R_l\right)^{-1} R_l^t \widetilde{D}^{-1}, \tag{27}$$

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

$$v = M_{\text{add}}^{-1} r = \hat{M}^{-1} \left[I - \left(\widetilde{J}^D - \hat{M}\right)(t_p + t_s)\right] r, \tag{28}$$

and

$$v = M_{\text{mult}}^{-1} r = \hat{M}^{-1} \left[I - \left(\widetilde{J}^D - \hat{M}\right)\left(t_p + t_s - t_s \widetilde{J} t_p\right)\right] r. \tag{29}$$

The difference between the additive and multiplicative preconditioners resides in the inclusion of the cross term $t_s \widetilde{J} t_p$ resulting from the computation of a new residual in step 6 of Algorithm 7.2.

8 Constraint operators

Constraint operators are primarily used to solve the original problem into a smaller dimensional space. In order to decrease the computational requirements of the preconditioners described above we use constraint operators. These operators are the basis for many well known projection operators such as: deflation, semicoarsening and supercoarsening multigrid operators.

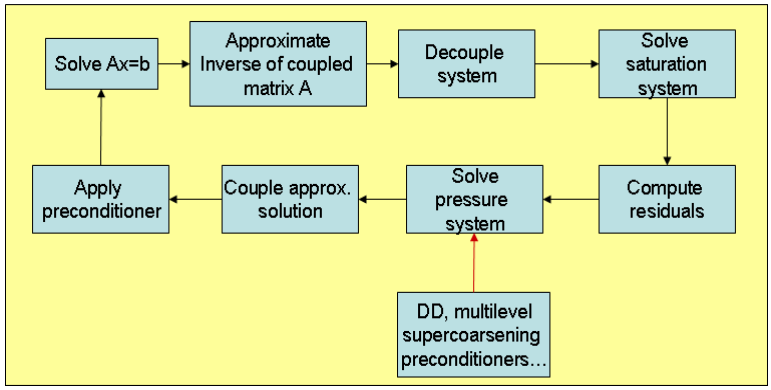

Figure 2: Schematic representation of two-stage preconditioning for solving coupled systems.

In the case of semicoarsening, 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 grid blocks along this direction and the plane coordinates as i and j with nx and ny grid blocks, respectively. Suppose a lexicographic numbering of nb unknowns with each one of them occupying 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

$$(C)_{lm} = \begin{cases} 1 & \text{if} \quad l = i + (j-1) * nx \text{ and } \; m = l + (k-1) * nx * ny, \\ 0 & \text{otherwise} \end{cases}$$

The semicoarsening can be described by the following steps for solving, say, the three dimensional algebraic system $Ty = r$:

Algorithm 8.1 (Semicoarsening)

1. Collapse residuals by means of the operator C^t and solve $\left(C^t T C\right) \widetilde{w} = C^t r$,

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 now proceed to describe how to incorporate this method in the framework of two-stage preconditioners. Consider the stage of 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 5.5.1,

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

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

Note that none of the operators $\widetilde{R}_p$, R_p or C have to be explicitly formed for implementation purposes. Moreover, approximation of the reduced matrices in pressure or concentration 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 implicitly performed onto the residuals to be preconditioned.

8.1 Supercoarsening

The supercoarsening multigrid (SCMG) method consist of a x-line relaxation (LSOR sweeps) and the yz-plane correction T. The correction implies solution of 2D aggregated system of equations in yz-plane. This is performed by a 2D multigrid with "two by two" coarsening and powerful relaxations on every level. The SCMG method features:

- Very low arithmetical complexity,

- Almost zero sum in vertical lines for the residual. It is good if the number of horizontal mesh layers is not very big,

- Convergence dependence on the number of mesh steps in horizontal direction
- Sensitivity (not very strong) to permeabilities,
- Very strong reduction of unknowns due to the first stage aggregation.

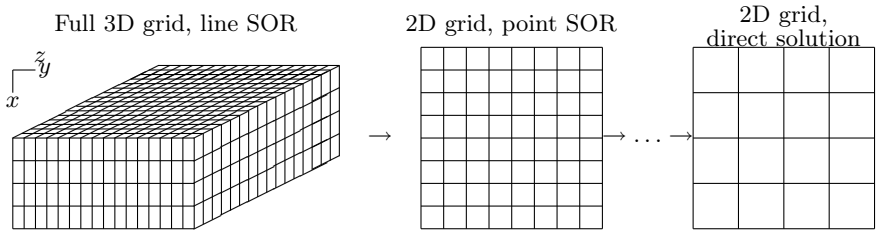

Figure 3: *Super-coarsening multigrid preconditioner.*

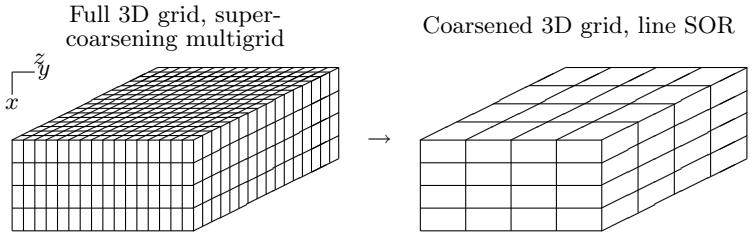

Figure 4: *Additional correction for the super-coarsening multigrid preconditioner.*

Further details of the SCMG method can be seen in [13, 16, 17].

9 Numerical Experiments

The first case involves a simplification of Richards' equation, which is used to model groundwater transport in the unsaturated zone. This time-dependent model in two space dimensions serves as a window to observe the HONK algorithms in action for subsurface simulation applications. We believe that this (or a similar) algorithm should benefit reservoir simulations in use by the petroleum and environmental industries.

9.1 Richards' equation

This example problem models that infiltration of the near surface underground zone in a vertical cross-section. This is a case of unsaturated flow that takes place in the region between the ground surface and the water table, i.e., the so called *vadose zone* the flow is driven by gravity and the transport coefficients are modelled by empirical nonlinear functions of the moisture content below saturation conditions. Thus, the model equation for this two-dimensional flow is given by

$$\frac{\partial c}{\partial t} - \nabla \cdot \left[D\left(c\right) \cdot \nabla c \right] - \frac{\partial K\left(c\right)}{\partial z} = 0,$$

where c is the underground moisture content, $D\left(c\right)$ is the dispersivity coefficient and $K\left(c\right)$ is the hydraulic conductivity. This equation is a simplification of the well known Richards' equation which also presents nonlinearities in the transient term.

The boundary conditions for this model are of Dirichlet type at the surface, where a constant water content of unity is kept at all times, and zero Neumann on the remaining three sides of the model. These conditions are given by

$$\begin{aligned} c = c_s, \quad \text{at} \quad z = 0, \quad 0 < y < 1, \\ \frac{\partial c}{\partial z} = 0, \quad \text{at} \quad z = 1, \quad 0 < y < 1, \\ \frac{\partial c}{\partial y} = 0, \quad \text{at} \quad y = 0 \quad \text{and} \quad y = 1, \quad 0 < z < 1, \end{aligned}$$

for $t > 0$. Here, z represents the vertical direction and y represents the chosen horizontal direction for the cross-section. The surface water content is represented by $c_s = 1$.

It would be appropriate to say that the same hydraulic conductivity plays a role in both the diffusive and the convective terms, because this model is just the continuity equation for the moisture content, where Darcy's law (with a moisture content dependent hydraulic conductivity) has been replaced for the superficial velocity. In fact, $D(c)$ is often referred to as the capillary diffusivity and is given by

$$D(c) = \frac{K(c)}{-\frac{dc}{d\psi}}, \quad \text{with} \quad \psi = -\frac{p}{\rho},$$

where p and ρ are the groundwater capillary head and density, respectively. However, different functional forms are often used to describe the dependence

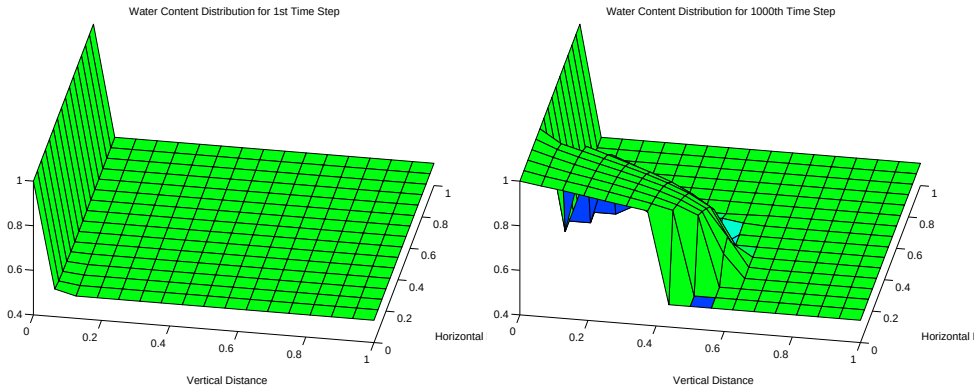

Figure 5: Water content distribution at 1^{st} and 1000^{th} time steps.

of both coefficients on the subsurface water content. For this example, our choices of dispersivity and hydraulic conductivity are, respectively,

$$D(c) = K_0 c_e^{\frac{1}{2}} \left[1 - \left(1 - c_e^2\right)^{\frac{1}{2}}\right]^2, \quad \text{and} \quad K(c) = K_0 c_e^{\frac{1}{2}},$$

with

$$c_e = \frac{c - c_0}{c_s - c_0},$$

where c_0 is the irreducible underground water content and the tensor K_0 is a position dependent coefficient whose nonzero values have been chosen according to

$$K_0(i,j) = \frac{5}{\mid i - j \mid +1}, \quad \text{for} \quad 1 \leq i, j \leq 5.$$

This choice of K_0, although contrived, is sometimes found in underground formations and, represents a narrow channel of permeable rock where the moisture is allowed to move. The hydraulic conductivity at saturation is proportional to the rock permeability, which has been shown to change over a few orders of magnitude within relatively short distances in underground formations. In our computational experiments we take $c_0 = 0.25$, which represents a typical value of the irreducible water content. See [2] for a comprehensive discussion of this model.

Figure 5 shows the solution for the moisture content distribution over the two-dimensional domain for a mesh of 16×16 at the 1^{st} and 1000^{th} time steps of simulation. The solution shows the effect of the heterogeneity in the resulting subsurface water content.

A constant time step was used for these simulations, which was chosen small enough to allow the inexact Newton method to converge within 40 nonlinear iterations and given by

$$\Delta t = \frac{1}{16} h^2.$$

This small time step was required in order to use the solution of the previous time level as an acceptable initial guess for the nonlinear iteration.

Figure 6 shows the accumulated (million) floating point operations for all the nonlinear methods as the simulation progresses up to 100 time steps, for a discretization mesh of 32×32. No preconditioning was used. The curve clearly exhibits the computational cost trend of all nonlinear methods.

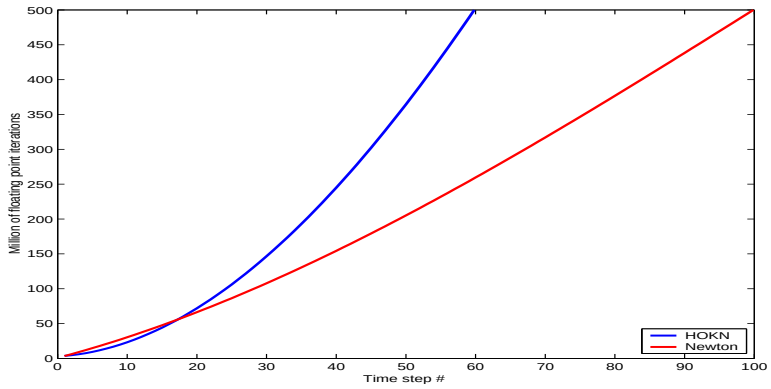

Figure 6: Performance in accumulated millions of floating point operations of Newton's method and the HONK method.

The HONK algorithm shows a significant saving in computational cost from start to end of this short simulation (as depicted in Figure 6). The increasing difficulty of the nonlinear problems as simulation advances produces a superlinear growth in the number of floating point operations. This growth is not only due to the complexity of the nonlinear problems but also to that of the linear problem. This is an example where the region of rapid convergence is far from the initial guess given at every time step, causing unexpected difficulties to Newton's method before reaching that region. It can be observed, for this particular case, that secant methods produce more efficient steps toward the solution, making them more preferable than Newton type of approaches.

Table 1 shows a comparative convergence behavior of a suite of several methods. The table confirms the excessive work (in terms of nonlinear itera-

Table 1: Total of nonlinear iterations (NI) and GMRES iterations (GI) for inexact versions of several nonlinear solvers. The problem size considered is of size 16×16 gridblocks after 100 time steps of simulation.

Method	NI	GI
Newton	1627	11890
Comp. Newton	835	12186
HONK	391	2673
Broyden	631	4046
NEN	347	4400
KEN	422	2757

tions) carried out by Newton's method and the composite Newton's method (also known as Shamanskii method, a higher-order modification of Newton's method, see [11]) compared to the HONK algorithm and all secant methods. The composite Newton's method halves the number of nonlinear iterations of Newton's method but both spend about the same total number of linear iterations. The HONK method reduces in half the number of nonlinear iterations taken by the composite Newton's method and, besides, it reduces by an almost 4-fold the total number of linear iterations with respect to this high-order Newton method. Hence, the HONK algorithm not only tackles efficiently the nonlinearities but also leads to much easier linear problems that arise at the neighborhood of the solution.

The NEN algorithm also halves the number of nonlinear iterations shown by Broyden's method but the number of linear iterations accounts for the similar computational efficiency of both. The nonlinear KEN algorithm takes an intermediate number of nonlinear iterations between these two algorithms but its efficiency is marked by the fewer number of displayed linear iterations.

9.2 Air-water model

We consider the 2-D Vauclin problem. This problem was implemented by Jenkins [10] for validating the air-water model in IPARS (Integrated Parallel Accurate Reservoir Simulator) [18].

The dimensions of the problem are $20ft \times 400ft \times 20ft$, with corresponding number of gridblocks $10 \times 20 \times 1$ along Z, Y and X; that is, a 2-D case in the plane $Z - -Y$. Initial pressures and saturations are computed at equilibrium from a pressure value of 503.94511 psi and a saturation of .5.

In order simulate evaporation and infiltration processes accordingly, air

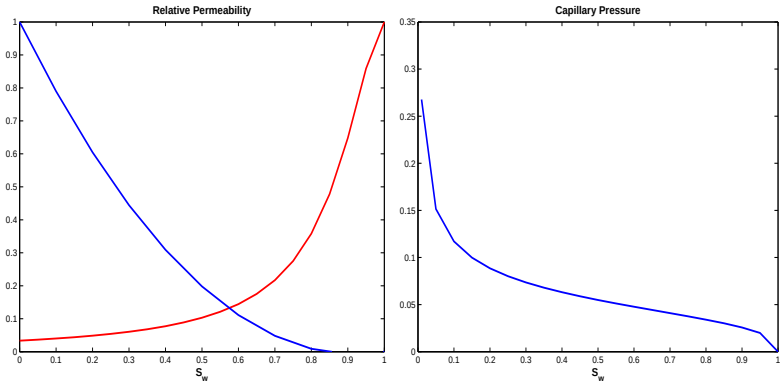

Figure 7: Saturation curves for the Vauclin problem.

and water production wells are placed in one extreme of the domain and one water injection well at the opposite end. Bottom hole pressures are specified to maintain a constant pressure in the system after 10 days of simulation. The system has an anisotropic definition of permeability with higher values at the side of the ditch (where the pumping/evaporation takes place). Porosity values are held constant to 20% everywhere. Saturation curve data are shown in Fig. 7. For representing the pressure head at the ditch, the capillary pressure curve shown at the right of Fig. 7 is decreased by 2 orders of magnitude.

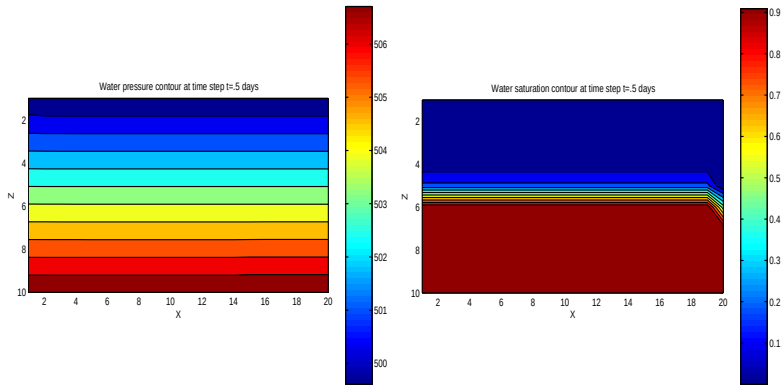

Figure 8: Pressure (left) and saturation (right) contour of the Vauclin problem at $t = .5$ days.

Figure 8 shows the field pressure and field saturation values at the beginning of the simulation. Figure 9 shows the resulting pressure and saturation

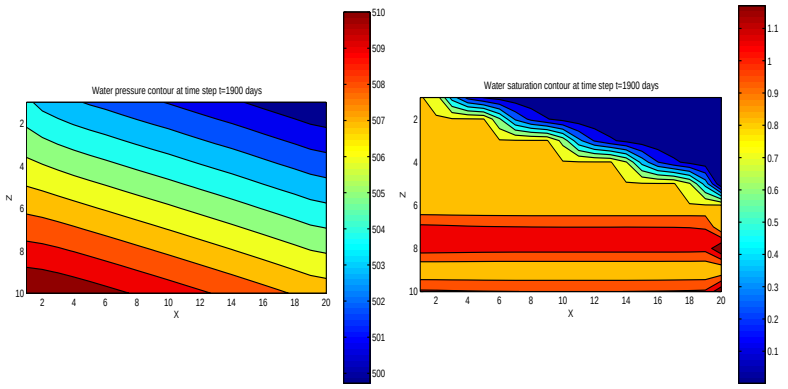

Figure 9: Pressure (left) and saturation (right) contour of the Vauclin problem at $t = 1900$ days.

field after 1900 days of simulation. We can clearly observed that the recharge area is almost linear due to the pumping effect caused by the well located at the right side of the reservoir. As expected pressures are higher at the left bottom extreme due to the infiltration zone in the lower permeability zone.

The associated linear systems (matrices and right hand side vectors) for the Vauclin test problem were generated by the two-phase option of IPARS. The data for the tests were downloaded from the simulation after 1 time step (0.5 days) and after 3064 time steps (1900 days). Implementation of the different preconditioning options were performed in MATLAB v6.1 on a Linux based PC machine.

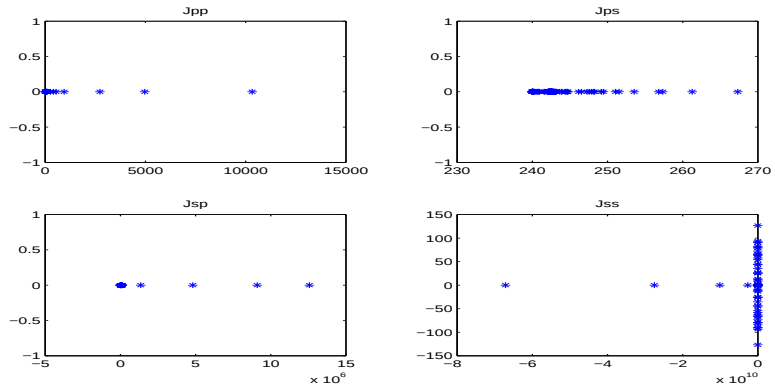

Figure 10: Spectra of the blocks composing the sample Jacobian matrix from the air-water case.

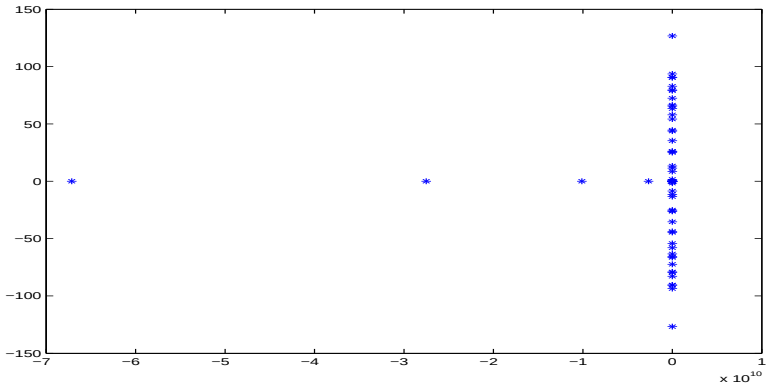

Figure 11: Spectrum of the sample Jacobian matrix from the air-water case

Figure 10 shows the spectrum of each of the blocks involved in the Jacobian matrix obtained in the first nonlinear iteration at $t = 1900$ days. Looking at Fig. 11 we can verify the resemblance between the spectrum of J_{ss} and that of the Jacobian matrix J. Therefore, the conditioning of block J_{ss} has an immediate incidence on the conditioning of the whole system. In this respect, we can expect the decoupling operator D^{-1} to do a better preconditioning job than the W_p operator (see discussion at the end of section 6). Moreover, the resulting decoupled matrix is weak block diagonally dominant (i.e., the spectral radius is bounded by one) and therefore classical block iterative methods converge for this matrix for any right hand side and initial guess [13]. We also note that despite the small size of the problem, the condition number for the Jacobian matrix J is very high. Also, J is highly indefinite.

In Figure 12 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 by the action of D^{-1}. In contrast, strategies that intent to preserve as much as possible 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 unknowns) than trying to preserve some desirable properties of the individual blocks.

Figure 13 (top) shows the spectrum of the combinative operator applied to the Jacobian matrix for an exact solution of the pressure system. In this particular example, $\hat{M}$ was taken to be the diagonal (i.e. jacobi precondi-

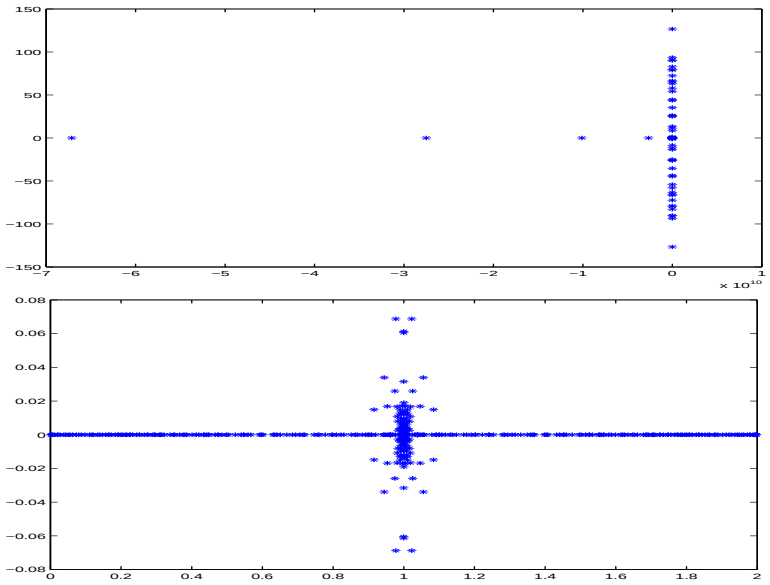

Figure 12: Spectrum of the decoupled sample Jacobian matrix from the air-water case: Applying the operator W_p (top) and D^{-1} (bottom).

tioner) part of J. For the additive and multiplicative two-stage preconditioner, we specify $\hat{M}$ to be the identity matrix. The preconditioner does a good job in clustering the eigenvalues, although we can predict that there are some small ones that will affect the convergence of an Krylov iterative solver. In contrast, Fig. 13 also shows how the two-stage additive (middle) and two-stage multiplicative (bottom) perform a more efficient task in clustering the spectrum around the point $(1, 0)$ in 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 two-stage preconditioners.

Looking at the convergence of GMRES with different preconditioners some general comments are in order (see Table 2). Linear tolerance was set to machine precision: 1.10^{-17}. 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 two-stage preconditioners. In particular ILU(0) and block Jacobi do not make GMRES to converge at the required precision within 400 iterations. These two preconditioners present difficulties to resolve the low error frequencies of the solution. We do not include them in the table.

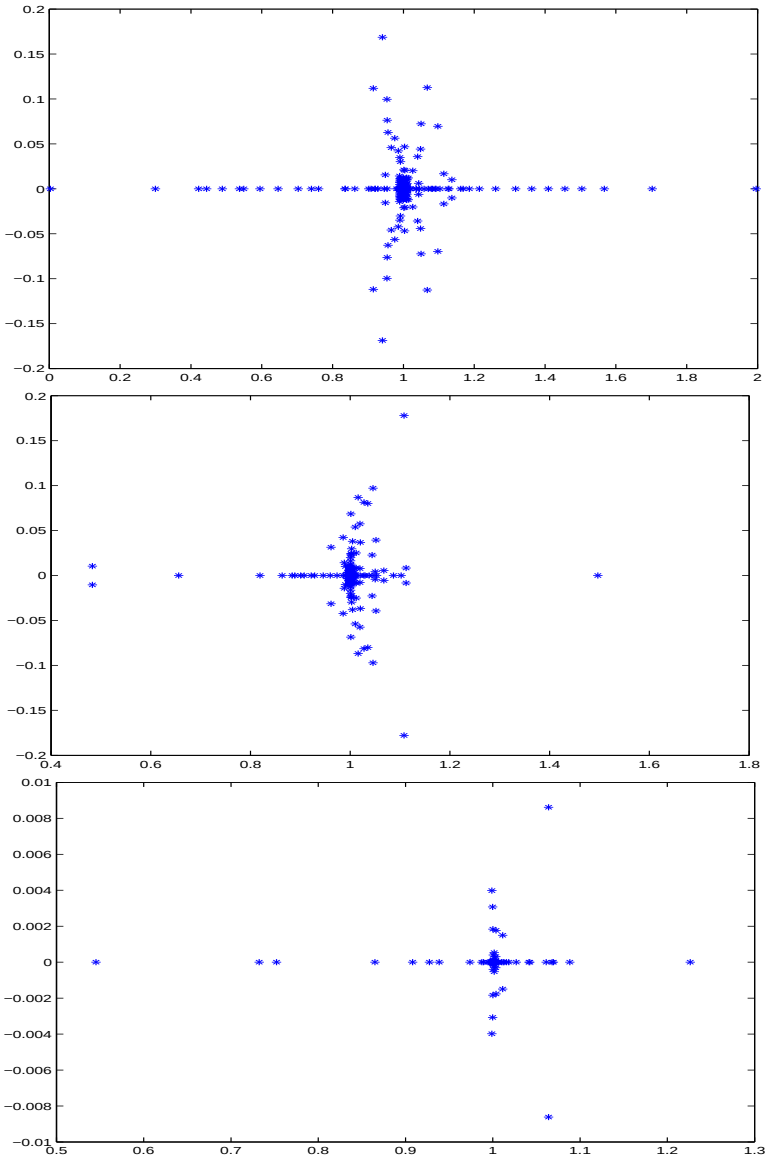

Figure 13: Spectrum of the Jacobian matrix preconditioned with the M_{comb} operator (top), the M_{add} operator (middle) and M_{mult} operator (bottom).

As was expected from the spectrum information, the two-stage multiplicative preconditioner is the most effective one. This trend was observed for different simulation times and time step sizes. The additive and multiplicative version outperformed the combinative one in all cases and more noticeable performance differences were observed as the time step was increased.

However, all of the two-stage (except for the 2SComb) preconditioners give a smaller number of outer iterations for the longer time step. The key in interpreting these results is in the action of the full-decoupling operator implemented for the two-stage preconditioners and its own power to precondition the system. We believe that the *weight* of the off-diagonal Jacobian blocks after full decoupling is less for the longer time step than for the shorter one and the preconditioner is more effective as a result.

Table 2: Total of linear iterations for GMRES with different two-stage preconditioners.

Preconditioner	$t = .5$ days	$t = 1900$ days
Combinative	20	119
Additive	14	24
Multiplicative	13	16

10 Conclusions and Further Work

Two novel Newton-Krylov algorithms were proposed for the inexact solution of nonlinear systems of algebraic equations with potential for higher-order convergence when applied to a variety of problems. Both of these methods are based on implicit Krylov-secant updates of the Jacobian matrix (i.e., Broyden updates of the Hessenberg matrix resulting from the Arnoldi decomposition in GMRES). The nonlinear KEN algorithm is an inexact form of that proposed recently by Yang with the added efficiency of Broyden updates on the lower-dimensional Hessenberg matrix. This method shows a computational performance which is far superior to that of either Broyden's or the NEN algorithms. The HONK method uses the Krylov-Broyden updates to generate a series of chord steps restricted to the Krylov subspace, which results in an extremely efficient secant version of the inexact composite Newton implementation (potentially with order higher than 2). This algorithm outperforms both the Newton and standard composite Newton

method in terms of floating point operations required for convergence to the same tolerance.

The possibilities for extending both methods are enormous. First, more robust versions can be developed by enriching the underlying Krylov information. This can be realized by augmenting, orthogonalizing and deflating (preconditioning) the Krylov basis. Advances along these lines have been useful to develop Krylov subspace method for solving linear systems with different right hand sides and a changing sequence of linear systems in both matrix and the right hand side [20, 22].

A second source of improvement has to do with the projection of Krylov information in successive time steps. The idea will be to obtain a better initial guess for the nonlinear method. This will also imply an increase of the region of convergence and a further accelerating scope for both the KEN and HONK methods. Attempts to generate stable time stepping from Krylov basis information has been reported in [3].

Our numerical results show that the new preconditioners outperform a few traditional approaches proposed in the literature of environmental and reservoir engineering: ILU(0) and block Jacobi preconditioner (for preconditioning of the entire system), and an inexact version of the combinative approach (originally developed for coupled linear systems).

We also discussed, in a algebraic fashion, the use of constraint operators for performing semi- and supercoarsening multigrid methods. The use of these operators are effective whenever the contrast of coefficients is not severe along the direction where the coarsening is performed. Nevertheless, the success of these methods will depend upon the interplay that can be built in deflating eigenvalues between the Krylov iterative method and the projection of the problem into smaller subproblems.

We consider that the following issues on two-stage preconditioners and constraint operators need to be addressed in the future:

- Dynamic characterization of the tolerances controlling the inner component solvers of the preconditioner.

- Theoretical analysis and extension of the preconditioners to several unknowns per gridblock. It is important to know what choice of primary variables and what properties can be exploited when solving large coupled systems of nonlinear equations. We have observed that some decoupled blocks are easier to solve than others, so this may determine the type of linear solvers to be used within the preconditioner.

- Study the deflation properties of these methods and their complementarity with deflating Krylov iterative methods (see, e.g., [4, 19]).

The results obtained are a valuable insight for solving 3-D air-water problems with discontinuous Galerkin (DG) methods. Currently, our research group is studying two level domain decomposition approaches for solving pressures and local DG for saturations. The decoupling framework developed here serves to the separate treatment of both physical entities.

In the case of coupling flow and geomechanics, the linear system dramatically increases in complexity. Besides pressures and saturations, the linear system includes equations for the linearized poro-elasticity model and displacements need also to be solved. However, simple decoupling arguments as the ones described here with a judicious application of quadrature rules can reduce the 27-point stencil to a 7-point stencil. An algebraic multigrid algorithm is used to solve each displacement component. To reduce the CPU time consumed in the setup phase of AMG, we have proposed to use AMG for solving the residual corrections in an aggregated subspace. According to preliminary numerical experiments, the resulting preconditioners are very robust with respect to both increasing Poisson ratio and large jumps in coefficients. It's convergence is fairly stable with respect to mesh parameter h. High-order Newton-Krylov method should also aid at reducing the turnaround time.

In general, further computational experiences are required. Our results are promising but need to be evaluated under more stringent physical situations and at a larger scale. Among target applications we consider useful the insights coming from three-phase, compositional and their coupling to geomechanics and chemical processes. The different strengths in nonlinearities between the typical coupled equations of the subsurface model (i.e., one for the pressure potential or flow driving force and another or others for the saturations/ concentrations) and the coupling itself between reservoir variables are sufficiently good reasons to investigate more on the preconditioning ideas displayed in this project.

Acknowledgements

The authors DOD-PET Agency for its financial support through the project EQM04-003, grant N62306-01-D-7110.

References

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

[2] J. Bear. *Dynamics of Fluids in Porous Media*. Dover Publications, Inc, 1972.

[3] M. Botchev, G.L.G. Sleijpen, and H. van der Vorst. Stability control for approximate implicit time stepping schemes with minimum residual iterations. *Applied Numerical Mathematics*, 31:239–253, 1999.

[4] K. Burrage, J. Erhel, B. Pohl, and A. Williams. A deflation technique for linear system of equations. *SIAM J. Sci. Statist. Comput.*, 19(4):1245–1260, 1998.

[5] C. Dawson, H.M. Klíe, M.F. Wheeler, and C. Woodward. A Parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver. *Comp. Geosciences*, 1:215–249, 1997.

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

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

[8] T. Eirola and O. Nevanlinna. Accelerating with rank–one updates. *Linear Alg. and its Appl.*, 121:511–520, 1989.

[9] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.

[10] E. W. Jenkins. The IPARSv2 air-water model. Technical Report TICAM report 02-27, The University of Texas at Austin, ICES, 2002. http://www.ices.utexas.edu/reports/2002.html.

[11] C.T. Kelley. Iterative methods for linear and nonlinear equations. In *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1995.

[12] H. Klie. *Krylov-Secant methods for solving Large Scale Systems of Coupled Nonlinear Parabolic Equations*. PhD thesis, Dept. of Computational and Applied Mathematics , Rice University, Houston, TX, 1996.

[13] H. Klie and Mary F. Wheeler. Constraint and decoupling operators for preconditioning implicit two-phase model formulation of groundwater flow problems. Technical Report TICAM Report (in preparation), The University of Texas at Austin, 2004. To be available at http://www.ices.utexas.edu/reports/2004/index.html.

[14] H. Klie and Mary F. Wheeler. High order Newton-Krylov methods for solving Richards' equation. Technical Report TICAM Report (in preparation), The University of Texas at Austin, 2004. To be available at http://www.ices.utexas.edu/reports/2004/index.html.

[15] D. A. Knoll and D. E. Keyes. Jacobian free Newton-Krylov methods: A survey of approaches and Applications. *Journal of Computational Physics*, 20(2):357–397, 2004.

[16] S. Lacroix, Yu. Vassileski, J. Wheeler, and M.F. Wheeler. Iterative solution methods for modeling multiphase flow in porous media fully implicitly. Accepted for publication in SIAM J. Sci. Comput., 2004.

[17] S. Lacroix, Yu. Vassileski, and Mary F. Wheeler. Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS): Single processor case. Technical Report TICAM Report 00-28, The University of Texas at Austin, 2000. http://www.ices.utexas.edu/reports/2000/index.html.

[18] Q. Lu, M. Peszynska, and X. Gai. Implicit Black-Oil model in IPARS framework. Technical Report TICAM report 01-34, The University of Texas at Austin, ICES, 2001. http://www.ices.utexas.edu/reports/2001.html.

[19] R. B. Morgan. GMRES with deflated restarting. *SIAM J. Sci. Statist. Comput.*, 24(1):20–27, 2002.

[20] M.L. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. Technical Report Technical Report UIUCDCS-R-2004-2421 (CS), UILU-ENG-2004-1722 (ENGR), Dept. of Computer Science, University of Illinois at Urbana-Champaign, 2004.

[21] M. Pernice and H.F. Walker. NITSOL: A Newton iterative solver for nonlinear systems. *SIAM J. Sci. Comput.*, 19:302–318, 1998.

[22] F. Risler and C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical Algorithms*, 23:1–30, 2000.

[23] H. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Camdbrige University Press, 2003.

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

[25] U.M. Yang. *A family of preconditioned iterative solvers for sparse linear systems*. PhD thesis, Dept. of Computer Science, University of Illinois, Urbana-Champaign, 1995.