INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:1329–1346
Published online 1 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1025

Projection and partitioned solution for two-phase flow problems

Andrea Comerlati*,†, Giorgio Pini‡ and Giuseppe Gambolati§

Department of Mathematical Methods and Models for Scientific Applications, University of Padova, via Belzoni 7, 35131 Padova, Italy

SUMMARY

Multiphase flow through porous media is a highly nonlinear process that can be solved numerically with the aid of finite elements (FE) in space and finite differences (FD) in time. For an accurate solution much refined FE grids are generally required with the major computational effort consisting of the resolution to the nonlinearity frequently obtained with the classical Picard linearization approach. The efficiency of the repeated solution to the linear systems within each individual time step represents the key to improve the performance of a multiphase flow simulator. The present paper discusses the performance of the projection solvers (GMRES with restart, TFQMR, and BiCGSTAB) for two global schemes based on a different nodal ordering of the unknowns (ORD1 and ORD2) and a scheme (SPLIT) based on the straightforward inversion of the lumped mass matrix which allows for the preliminary elimination and substitution of the unknown saturations. It is shown that SPLIT is between two and three time faster than ORD1 and ORD2, irrespective of the solver used. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: finite elements; two-phase flow in porous media; projection methods; partitioned procedure; direct solvers

1. INTRODUCTION

Immiscible two-phase flow in porous media in isothermal conditions is described by the mass conservation equation of each phase [1, 2]:

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} = -\nabla\cdot[\rho_\alpha \mathbf{v}_\alpha] + \mathbf{q}_\alpha, \quad \alpha = \mathrm{w}, \mathrm{n} \tag{1}$$

*Correspondence to: Andrea Comerlati, Department of Mathematical Methods and Models for Scientific Applications, University of Padova, via Belzoni 7, 35131 Padova, Italy.
†E-mail: andreac@dmsa.unipd.it
‡E-mail: pini@dmsa.unipd.it
§E-mail: gambo@dmsa.unipd.it

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389
Contract/grant sponsor: PRIN-MIUR

Received 25 February 2005
Revised 9 May 2005
Accepted 10 May 2005

where subscript α refers to wetting (w) and non-wetting (n) phase, respectively (e.g. water and oil or water and gas). S_α is the saturation, ρ_α the density, $\mathbf{v}_\alpha$ the Darcy velocity, $\mathbf{q}_\alpha$ the mass source/sink rate of the phase α, and ϕ denotes the porous medium porosity. The velocity of each phase α is given by the extended Darcy law:

$$\mathbf{v}_\alpha = -\lambda_\alpha \underline{\underline{k}}(\nabla p_\alpha - \rho_\alpha \mathbf{g}) \tag{2}$$

where the mobility λ_α is defined as the ratio between the relative permeability $k_{r\alpha}$ and the dynamic viscosity μ_α of the phase α, $\underline{\underline{k}}$ is the medium intrinsic permeability tensor, p_α the α-phase pressure, and $\mathbf{g}$ the gravity acceleration vector. Substituting Equation (2) into the continuity equations (1) yields:

$$\frac{\partial(\phi \rho_\alpha S_\alpha)}{\partial t} = \nabla \cdot [\rho_\alpha \lambda_\alpha \underline{\underline{k}}(\nabla p_\alpha - \rho_\alpha \mathbf{g})] + \mathbf{q}_\alpha \tag{3}$$

The solution of the PDEs system (3) requires the following auxiliary relationships:

$$S_w + S_n = 1, \quad p_c(S_w) = p_n - p_w \tag{4}$$

where S_w and S_n are wetting and non-wetting saturations and p_c the capillary pressure defined as the difference between the non-wetting and wetting-phase pressure. Capillary properties can be described using a number of constitutive laws, such as for instance the Brooks–Corey capillary model [3]:

$$k_{rw}(S_w) = S_{we}^{(2+3\zeta)/\zeta}, \quad k_{rn}(S_w) = (1 - S_{we})^2(1 - S_{we}^{(2+\zeta)/\zeta}), \quad p_c(S_w) = p_d S_{we}^{-1/\zeta}$$

where p_d is the pore entry pressure representing the lowest capillary pressure needed to displace the wetting phase by the non-wetting phase in a fully saturated medium, ζ the so-called sorting factor or pore distribution index which is related to the medium pore size distribution. The sorting factor usually ranges between 0.2 (denoting a wide range of pore sizes) and 7 (for very uniform materials), $S_{we} = (S_w - S_{wr})/(1 - S_{wr})$ is the effective water saturation, with S_{wr} the irreducible water saturation.

Using the auxiliary relationships (4), PDEs (3) can be rewritten in terms of water pressure (p_w) and water saturation (S_w) yielding the coupled pressure–saturation formulation:

$$\begin{aligned} \frac{\partial(\phi \rho_w S_w)}{\partial t} &= \nabla \cdot [\rho_w \lambda_w \underline{\underline{k}}(\nabla p_w - \rho_w \mathbf{g})] + \mathbf{q}_w \\ \frac{\partial(\phi \rho_n (1 - S_w))}{\partial t} &= \nabla \cdot [\rho_n \lambda_n \underline{\underline{k}}(\nabla p_w + \nabla p_c - \rho_n \mathbf{g})] + \mathbf{q}_n \end{aligned} \tag{5}$$

Equations (5) represent a highly nonlinear system of PDEs, where capillary pressure and relative permeability depend on saturation:

$$k_{rw} = k_{rw}(S_w), \quad k_{rn} = k_{rn}(S_w), \quad p_c = p_c(S_w)$$

while fluid density and viscosity may depend on the corresponding phase pressure:

$$\rho_w = \rho_w(p_w), \quad \rho_n = \rho_n(p_n), \quad \mu_w = \mu_w(p_w), \quad \mu_n = \mu_n(p_n)$$

As an example, the capillary pressure and relative permeability are provided in Figure 1 for the two-phase trichloroethylene (TCE)–groundwater system in a coarse sand.

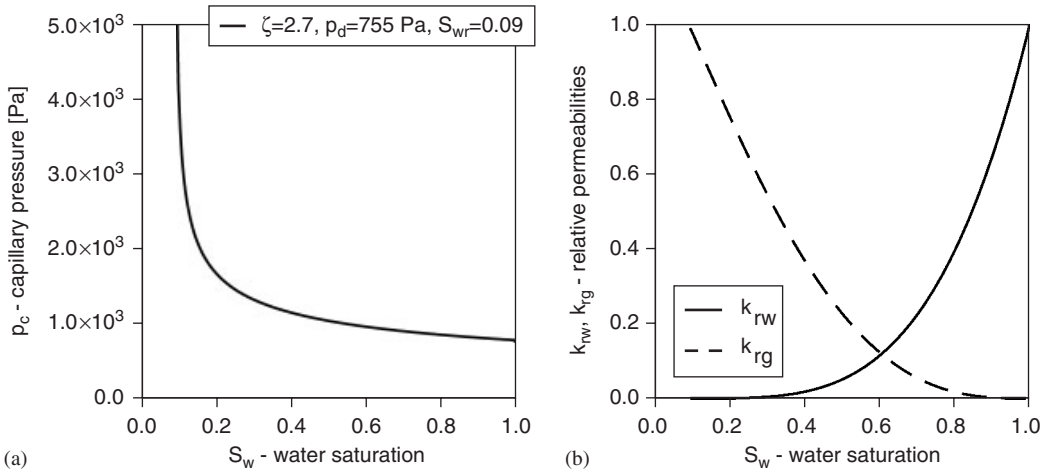

Figure 1. Brooks–Corey capillary pressure (a) and relative permeability (b) curves for a coarse sand, characterized by a sorting factor and an entry pressure $\zeta = 2.7$ and $p_d = 755\,\text{Pa}$, respectively. Irreducible water saturation is $S_{wr} = 0.09$.

2. TWO-PHASE FLOW FINITE ELEMENT EQUATIONS

Equations (3) are discretized in space using linear tetrahedra FE in 3D yielding a system of first-order differential equations that read (see Appendix A):

$$\begin{bmatrix} H_w & 0 \\ H_n & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_w \\ \boldsymbol{S}_w \end{bmatrix} + \begin{bmatrix} 0 & M_w \\ 0 & M_n \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_w \\ \dot{\boldsymbol{S}}_w \end{bmatrix} + \begin{bmatrix} \boldsymbol{q}_w \\ \boldsymbol{q}_n \end{bmatrix} = 0 \tag{6}$$

where H_w, H_n, M_w, and M_n are the wetting and non-wetting stiffness and mass matrices; $[\boldsymbol{q}_w, \boldsymbol{q}_n]^T$ incorporates source/sinks terms, gravity terms, and Neumann boundary conditions; $[\boldsymbol{p}_w, \boldsymbol{S}_w]^T$ and $[\dot{\boldsymbol{p}}_w, \dot{\boldsymbol{S}}_w]^T$ are the vectors of the unknown nodal water pressure ($\boldsymbol{p}_w$) and saturation ($\boldsymbol{S}_w$), and the corresponding time derivatives. Mass matrices M_w and M_n are lumped for stability reasons [4], while in the stiffness matrices H_w and H_n hydraulic mobility is evaluated 'fully upwind' [1, 2, 5–7] to ensure convergence of the nonlinear scheme to the correct physical solution and to avoid undesirable oscillations when capillary forces become small. Stiffness matrices H_w and H_n are symmetric positive definite and symmetric positive semi-definite, respectively. Mass matrices M_w and M_n are diagonal matrices. System (6) can be written in a more compact form as

$$H\boldsymbol{x} + M\dot{\boldsymbol{x}} + \boldsymbol{q} = 0 \tag{7}$$

where the meaning of the new symbols can be easily derived by comparison of Equations (6) and (7). The time integration is implemented via Euler backward FD, giving the following

nonlinear system of algebraic equations:

$$\left[H + \frac{M}{\Delta t}\right]_{(k+1)} \boldsymbol{x}_{(k+1)} = \left[\frac{M}{\Delta t}\right]_{(k+1)} \boldsymbol{x}_{(k)} - \boldsymbol{q}_{(k+1)} \tag{8}$$

where Δt is the time step size; (k) and $(k+1)$ indicate the previous and the current time level, respectively. The nonlinear system (8) is solved by Newton-like iterative methods. To this aim Equation (8) is rewritten as

$$f(\boldsymbol{x}_{(k+1)}) = A\boldsymbol{x}_{(k+1)} - \bar{\boldsymbol{q}} = 0$$

with

$$A = \left[H + \frac{M}{\Delta t}\right]_{(k+1)}, \qquad \bar{\boldsymbol{q}} = \left[\frac{M}{\Delta t}\right]_{(k+1)} \boldsymbol{x}_{(k)} - \boldsymbol{q}_{(k+1)}$$

Let $\boldsymbol{x} = \boldsymbol{x}_{(k+1)}$, expanding $f(\boldsymbol{x})$ in Taylor's series and denoting with (r) the nonlinear iteration counter, we obtain:

$$f(\boldsymbol{x}^{(r+1)}) = f(\boldsymbol{x}^{(r)}) + f'(\boldsymbol{x}^{(r)})(\boldsymbol{x}^{(r+1)} - \boldsymbol{x}^{(r)}) + \cdots = 0$$

setting the current search direction $\delta\boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r+1)} - \boldsymbol{x}^{(r)}$ and the derivative term $J(\boldsymbol{x}^{(r)}) = f'(\boldsymbol{x}^{(r)})$, the Newton iteration reads:

$$J(\boldsymbol{x}^{(r)})\delta\boldsymbol{x}^{(r+1)} = -f(\boldsymbol{x}^{(r)}), \quad \boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r)} + \delta\boldsymbol{x}^{(r+1)} \tag{9}$$

where the Jacobian matrix is given by $J = A + A'\boldsymbol{x} - \bar{\boldsymbol{q}}'$, the superscript indicates vector differentiation. When the Jacobian is approximated by neglecting A' and $\bar{\boldsymbol{q}}'$, the Picard scheme is obtained. In this case the linear system that must be solved at each nonlinear iteration can be written as

$$A(\boldsymbol{x}^{(r)})\delta\boldsymbol{x}^{(r+1)} = -A(\boldsymbol{x}^{(r)})\boldsymbol{x}^{(r)} + \bar{\boldsymbol{q}}(\boldsymbol{x}^{(r)}) \tag{10}$$

with the search direction given by $\delta\boldsymbol{x}^{(r+1)} = [\delta\boldsymbol{p}_{\mathrm{w}}^{(r+1)}, \delta\boldsymbol{S}_{\mathrm{w}}^{(r+1)}]^{\mathrm{T}}$. Nonlinear convergence is considered achieved when the norm of each correction component satisfies the following test:

$$|\delta\boldsymbol{p}_{\mathrm{w}}^{(r+1)}| < \tau_{\mathrm{prs}} \quad \text{and} \quad |\delta\boldsymbol{S}_{\mathrm{w}}^{(r+1)}| < \tau_{\mathrm{sat}} \tag{11}$$

where τ_{prs} and τ_{sat} are the nonlinear tolerances for water pressure and water saturation, respectively; $|\cdot|$ can either be the L_2 or the L_∞ norm. The nonlinear convergence properties of the Picard and Newton schemes are discussed in a number of References [8, 9]. In general one can say that the Picard scheme exhibits a good initial convergence properties, especially in combination with a relaxation technique. However, it often suffers from slow convergence or stagnation at relatively low residual levels. This stagnation can be related to the absence of the derivative terms in the Jacobian matrix. The behaviour of the Newton approach, on the other hand, is locally optimal, achieving quadratic convergence error at low residual levels. However, nonconvergence or even divergence may be observed at the initial stage of the iteration if the initial guess is not close enough to the final solution. Techniques to alleviate this problem include, for instance, relaxation or line search methods. These techniques may be used with both the Picard and the Newton approach in order to reduce the size of the step taken along the search direction $\delta\boldsymbol{x}$, leading to the update $\boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r)} + \omega\delta\boldsymbol{x}^{(r+1)}$. The

parameter $\omega \in]0,1]$ is called the relaxation or line-search parameter. The quality of the initial guess is crucial to obtain a fast nonlinear convergence and is influenced by the time step size. For this reason the value of Δt is empirically adapted on the basis of the convergence history at the previous time step, using the algorithm described below [10, 11]. The current Δt value is increased by a factor Δt_{mag} (up to a maximum $\Delta t_{\max}$) if convergence at the previous iteration is achieved in fewer than m_1 iterations, it is left unchanged if convergence requires between m_1 and m_2 iterations, and it is decreased by a factor Δt_{red} (down to a minimum $\Delta t_{\min}$) if convergence requires more than m_2 iterations. If convergence is not achieved (maximum number of iterations m exceeded), the iterative process is repeated (*back stepping*) using a reduced time step size (reduction factor Δt_{red}, down to $\Delta t_{\min}$). The Δt values and the maximum number of nonlinear iterations m, m_1, and m_2 are found empirically. In the present paper the Picard linearization procedure is used (see Equation (10)) on account of its simplicity ease of implementation and satisfactory computational performance experimented by various authors (e.g. References [12, 13]). The nonsymmetric system controlling the computational performance of the algorithm is given by

$$A^{(r)}\delta \boldsymbol{x}^{(r+1)} = \boldsymbol{b}^{(r)} \tag{12}$$

where matrix A and the right-hand side vector $\boldsymbol{b}$ can be written as

$$A^{(r)} = \begin{bmatrix} H_{\text{w}} & M_{\text{w}}/\Delta t \\ H_{\text{n}} & M_{\text{n}}/\Delta t \end{bmatrix}^{(r)}, \quad \boldsymbol{b}^{(r)} = -\begin{bmatrix} H_{\text{w}} & M_{\text{w}}/\Delta t \\ H_{\text{n}} & M_{\text{n}}/\Delta t \end{bmatrix}^{(r)} \begin{bmatrix} \boldsymbol{p}_{\text{w}} \\ \boldsymbol{S}_{\text{w}} \end{bmatrix}^{(r)} + \begin{bmatrix} \bar{\boldsymbol{q}}_{\text{w}} \\ \bar{\boldsymbol{q}}_{\text{n}} \end{bmatrix}^{(r)}$$

3. NUMERICAL RESULTS

Two different orderings (ORD1 and ORD2) and a partitioned solver (SPLIT) are tested for efficiency with a 3D example modified after Huber and Helmig [7]. In a homogeneous sandy sample 0.9×0.9 m large and 0.65 m high, initially saturated with water, a TCE infiltration takes place over a 0.1×0.1 m area located at the centre of the upper surface where a permanent TCE saturation of 0.25 is maintained for 300 s. The remaining part of the upper and the bottom boundaries are impermeable, whereas along the lateral sides hydrostatic water pressure is assumed. For symmetry reasons only a quarter of the sandy sample is discretized with a FE mesh consisting of 72 557 nodes and 404 352 tetrahedra (see Figure 2). Soil and fluid properties are summarized in Table I, and nonlinear and back-stepping parameters shown in Table II. All the numerical simulations are performed on a 32-bit PC-workstation equipped with a 1526 MHz AMD processor, 2000 Mbyte of core memory, and 256 kbyte of secondary cache. Only the comparison between iterative and direct solver has been performed also on a 64-bit Compaq workstation equipped with a 833 MHz Aplha EV6.8AL processor, 4500 Mbyte of core memory, and 8 Mbyte of secondary cache.

3.1. Solution by projection solvers

Projection (or conjugate gradient-like) methods essentially project the FE system onto subspaces (called Krylov subspaces) of increasing dimension ℓ and solve the projected system. The solution in the Krylov subspace is basically obtained by a minimal residual (MR) and an

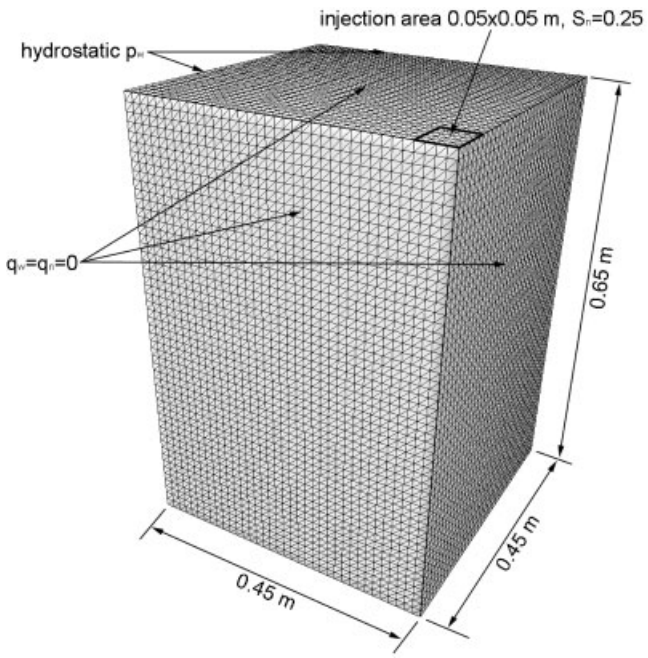

Figure 2. 3D mesh of a quarter of the core sample consisting of 72 557 nodes and 404 352 tetrahedra.

Table I. Soil and fluid properties.

Property	Symbol	Value
Intrinsic permeability (m^2)	k	6.64×10^{-11}
Porosity (dimensionless)	ϕ	0.40
Irreducible water saturation (dimensionless)	S_{wr}	0.09
Sorting factor (dimensionless)	ζ	2.7
Pore entry pressure (Pa)	p_d	755
Water density (kg/m^3)	ρ_w	1000
TCE density (kg/m^3)	ρ_n	1462
Water viscosity (Pa s)	μ_w	0.001
TCE viscosity (Pa s)	μ_n	0.00057

orthogonal residual (OR) procedure. The fundamental difference between MR and OR is that while with a MR method the existence of a solution is always theoretically guaranteed, the same does not necessarily hold with an OR method which might even diverge. The most attractive projection schemes for sparse nonsymmetric indefinite equations include the MR-based Generalized Minimum RESidual (GMRES) method algorithm [14], and the OR-based BiConjugate Gradient STABilized (BiCGSTAB) [15] and Transpose-Free Quasi-Minimal Residual (TFQMR) [16] methods. To be of a practical interest, iterative methods must be preconditioned. This implies transforming the original system into another system with the same

Table II. Tolerances and back-stepping parameters.

Parameter	Symbol	Value
Maximum # of nonlinear iterations (dimensionless)	m	8
Maximum # of nonlinear iterations (dimensionless)	m_1	6
Maximum # of nonlinear iterations (dimensionless)	m_2	8
Magnification factor (dimensionless)	Δt_{mag}	1.2
Reduction factor (dimensionless)	Δt_{red}	0.5
Maximum allowed time step (s)	Δt_{max}	10
Minimum allowed time step (s)	Δt_{min}	0.5
Simulated time (s)	t_{max}	300
Linear tolerance (dimensionless)	ε_l	10^{-5}
Nonlinear pressure tolerance (Pa)	τ_{prs}	10^{-2}
Nonlinear saturation tolerance (dimensionless)	τ_{sat}	10^{-4}

solution, but more cost-effective to solve. However, the construction of a preconditioner is not inexpensive. A good preconditioner realizes a most efficient trade-off between the opposite needs for reducing the preconditioner cost on the one hand, and accelerating the convergence of the preconditioned system on the other. An extensive review of the projection methods can be found in Reference [17].

On serial computers the most widely used preconditioner relies on the partial LU factorization of matrix A. For problems arising form the numerical integration of the subsurface flow equation the incomplete factorization with no fill-in ILU(0) [18, 19] turns out to work pretty well. Since its computation is cheap the preconditioner is updated and re-calculated within each nonlinear iteration. However, better results can be obtained with the incomplete factorization ILUT(ρ, τ) with variable fill-in connection with a suitable threshold strategy for dropping the small elements [20]. In the notation above, ρ represents the degree of fill-in and τ the threshold tolerance. The choice of the optimal ρ and τ values has to be made empirically for any specific problem. The efficiency of a given preconditioning technique in the above class is also related to the matrix bandwidth. In the present paper two different orderings are addressed, namely ORD1 and ORD2. The former is given by

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta p_{\text{w},1} \\ \vdots \\ \delta p_{\text{w},N} \\ \delta S_{\text{w},1} \\ \vdots \\ \delta S_{\text{w},N} \end{bmatrix}$$

ORD1 leads to a four block partitioned nonsymmetric matrix with a large bandwidth as is shown in Figure 3. The block corresponding to the non-wetting K_{n} stiffness matrix has non-zero coefficients only in correspondence to those nodal connections where the water saturation degree $S_{\text{w}} \neq 1$, and hence the relative permeability $k_{\text{rn}} \neq 0$.

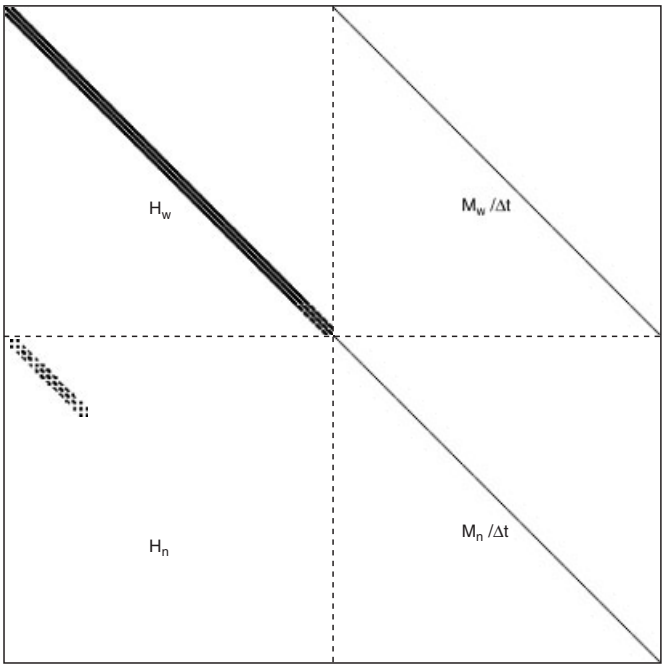

Figure 3. Structure of the system matrix arising from the unknowns ordering ORD1 after 300 s of TCE infiltration.

The second ordering, ORD2, interlaces pressure and saturation components as follows:

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta p_{\mathrm{w},1} \\ \delta S_{\mathrm{w},1} \\ \vdots \\ \delta p_{\mathrm{w},N} \\ \delta S_{\mathrm{w},N} \end{bmatrix}$$

ORD2 yields again a nonsymmetric matrix with a more reduced bandwidth (Figure 4) with the quality of the preconditioner expectedly improved and the convergence of the preconditioned projection schemes accelerated.

Convergence of the projection solver is considered achieved whenever the relative residual $r_{\mathrm{r}}^{(r,m)}$ meets the test below:

$$r_{\mathrm{r}}^{(r,m)} = \frac{|\boldsymbol{b} - A\delta \boldsymbol{x}^{(r+1,m)}|}{|\boldsymbol{b}|} < \varepsilon_l \tag{13}$$

where $|\cdot|$ is the L_2 norm, (m) the iteration counter, and ε_l the preset exit tolerance.

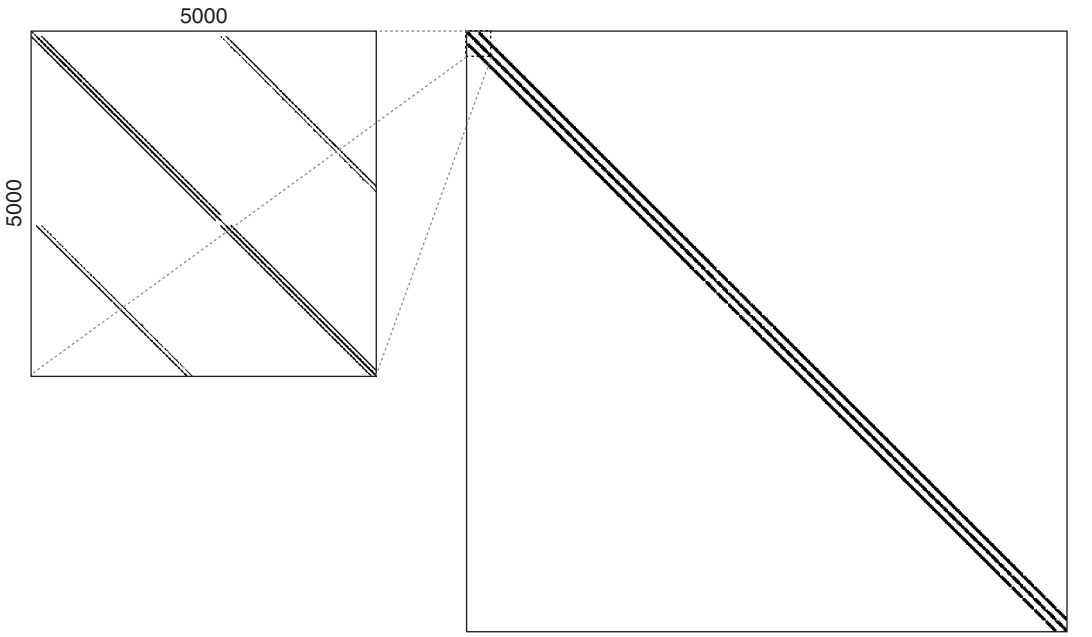

Figure 4. Structure of the system matrix arising from the unknowns ordering ORD2 after 300 s of TCE infiltration.

3.2. *Solution by partitioned solver* (*SPLIT*)

As was previously outlined mass matrices M_w and M_n are lumped diagonal and their inversion is computationally inexpensive [21]. Hence a new solution approach may be developed. To this aim set:

$$A_1 = H_w, \quad A_2 = H_n, \quad D_1 = M_w/\Delta t, \quad D_2 = M_n/\Delta t$$
$$\boldsymbol{b}_1 = \boldsymbol{q}_w, \quad \boldsymbol{b}_2 = \boldsymbol{q}_n, \quad \boldsymbol{x}_1 = \delta \boldsymbol{p}_w, \quad \boldsymbol{x}_2 = \delta \boldsymbol{S}_w$$

System (12) can be re-written as

$$A_1 \boldsymbol{x}_1 + D_1 \boldsymbol{x}_2 = \boldsymbol{b}_1 \tag{14}$$

$$A_2 \boldsymbol{x}_1 + D_2 \boldsymbol{x}_2 = \boldsymbol{b}_2 \tag{15}$$

from Equation (15) $\boldsymbol{x}_2$ is obtained as

$$\boldsymbol{x}_2 = D_2^{-1}(\boldsymbol{b}_2 - A_2 \boldsymbol{x}_1) \tag{16}$$

and substituted into Equation (14) to provide the half-reduced nonsymmetric linear system:

$$(A_1 - D_1 D_2^{-1} A_2)\boldsymbol{x}_1 = \boldsymbol{b}_1 - D_1 D_2^{-1} \boldsymbol{b}_2 \tag{17}$$

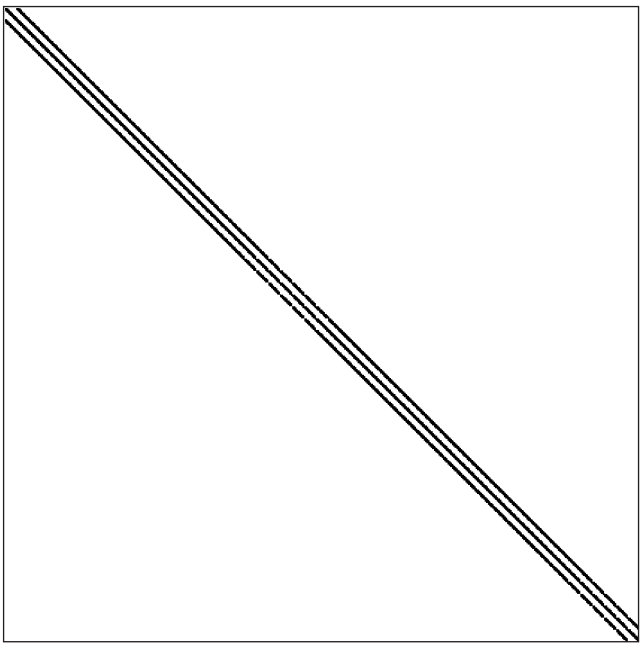

Figure 5. Structure of the system matrix arising from the SPLIT approach after 300s of TCE infiltration.

The structure of the matrix in Equation (17) is shown in Figure 5. The linear system is again solved by preconditioned projection methods and $\boldsymbol{x}_2$ finally calculated by replacing $\boldsymbol{x}_1$ in Equation (16).

When the wetting, non-wetting phases, and porous matrix are assumed to be incompressible, the product $D_1 D_2^{-1} A_2$ becomes:

$$D_1 D_2^{-1} A_2 = \frac{\rho_w}{\rho_n} A_2$$

System (17) can be efficiently solved with the symmetric preconditioned conjugate gradient with the incomplete Cholesky matrix factorization. By this approach much computer memory can be saved as only the upper triangular part of the system matrix given by $A_1 - \rho_w/\rho_n A_2$ must be stored in core.

3.3. Comparison between projection solvers

ORD1, ORD2, and SPLIT are discussed in connection with different projection methods. ILU(0) preconditioned GMRES, TFQMR, and BiCGSTAB schemes have been selected, with $(LU)^{-1}\boldsymbol{b}$ as the initial guess (with L and U the incomplete Crout factors).

The GMRES has been tested using $k = 10, 20, 40, 60$ as restart values. The best timing is obtained with $k = 20$ which appears to represent the rightest compromise between numerical efficiency and core memory occupation (see Table III). Switching from ORD1 to ORD2 the

Table III. Timing results after 300 s of TCE infiltration, using ILU(0) preconditioned GMRES(20).

	ORD1	ORD2	*SPLIT
Total linear system solution time (s)	8947	7327	3108
Total simulation time (s)	9495	7844	3616
Total # of time steps	67	67	67
Total # of back stepping	24	24	24
Total # of nonlinear iterations	530	530	530
Total # of linear iterations	23 045	18 781	23 848
Avg. linear iterations per time step	344	280	356
Avg. linear iterations per nonlinear iteration	43	35	45

*$N \times N$ linear system.

Table IV. Timing results after 300 s of TCE infiltration, using ILU(0) preconditioned TFQMR.

	ORD1	ORD2	*SPLIT
Total linear system solution time (s)	12 955	10 867	3863
Total simulation time (s)	13 512	11 382	4358
Total # of time steps	67	67	67
Total # of back stepping	24	24	24
Total # of nonlinear iterations	530	530	530
Total # of linear iterations	39 581	32 295	38 830
Avg. linear iterations per time step	591	482	579
Avg. linear iterations per nonlinear iteration	74	61	73

*$N \times N$ linear system.

computational cost is reduced by a factor 1.2. However, a more significant improvement is achieved with SPLIT which reduces the computational cost relative to ORD1 by a factor 2.6.

The TFQMR timing is given in Table IV, convergence being almost the same as GMRES. ORD1 and ORD2 provide a similar outcome, while a better performance is obtained with SPLIT whose computational cost relative to ORD1 is reduced by a factor 3.1. However, overall TFQMR requires a larger computational cost than GMRES with the most convenient restart value. For example TFQMR-SPLIT requires a total computational cost of 4358 s, resulting almost 20% slower than GMRES(20)-SPLIT.

Timing from BiCGSTAB is finally provided in Table V. Observe in Table V that the overall performance of the preconditioned BiCGSTAB is superior to both GMRES and TFQMR, for example relative to GMRES(20)-SPLIT, the total simulation time of BiCGSTAB-SPLIT is reduced from 3616 to 3143 s. BiCGSTAB appears to be quite robust and does not require the use of any problem dependent parameter.

3.4. Comparison with a direct solver

BiCGSTAB-SPLIT is compared for efficiency also with a direct solver. We employed a direct solver based on a sparse multifrontal variation of the Gauss elimination method as it is

Table V. Timing results after 300 s of TCE infiltration, using ILU(0) preconditioned BiCGSTAB.

	ORD1	ORD2	*SPLIT
Total linear system solution time (s)	8543	7151	2653
Total simulation time (s)	9100	7664	3143
Total # of time steps	67	67	67
Total # of back stepping	24	24	24
Total # of nonlinear iterations	530	530	530
Total # of linear iterations	13 952	11 487	14 531
Avg. linear iterations per time step	208	171	216
Avg. linear iterations per nonlinear iteration	26	22	27

*$N \times N$ linear system.

Table VI. Timing results after 300 s of TCE infiltration.

	BiCGSTAB-SPLIT	MA41-SPLIT
Total linear system solution time (s)	2140	108 339
Time per nonlinear iteration (s)	3.97	201
Total # of time steps	66	66
Total # of back stepping	25	25
Total # of nonlinear iterations	539	539
Total # of linear iterations	14 694	—
Avg. linear iterations per time step	223	—
Avg. linear iterations per nonlinear iteration	27	—

Simulations performed on a 64-bit Compaq workstation equipped with a 833 MHz EV6.8AL processor, 4500 Mbyte of core memory, and 8 Mbyte of secondary cache.
*$N \times N$ linear system.

implemented in the MA41 routine of the Harwell Software library (HSL) [22]. The comparison is performed using two different computers:

1. A 32-bit PC-workstation equipped with a 1526 MHz AMD processor, 2000 Mbyte of core memory, and 256 kbyte of secondary cache.
2. A 64-bit Compaq workstation equipped with a 833 MHz Alpha EV6.8AL processor, 4500 Mbyte of core memory, and 8 Mbyte of secondary cache.

This is done in order to investigate the sensitivity of the linear solver to the computer architecture. Achieved timing on the Compaq machine is shown in Table VI with the direct solver MA41-SPLIT that turns out to be almost 50 times slower than the iterative method BiCGSTAB-SPLIT. Even worse results are obtained with the 32-bit PC-workstation where MA41-SPLIT appears to be almost 420 times slower than BiCGSTAB-SPLIT. This is due to the different cache dimension which apparently has a great influence on the access to the data stored within the core memory. Switching from BiCGSTAB-SPLIT to MA41-SPLIT the memory requirement is increased from 160 to 1200 Mbyte, thus precluding the use of more refined computational grids.

One last point bears mention. Inequality (13), when recasted using the notation of the Newton scheme can be written as

$$|J(\boldsymbol{x}^{(r)})\delta\boldsymbol{x}^{(r+1)} + f(\boldsymbol{x}^{(r)})| \leqslant \varepsilon_l |f(\boldsymbol{x}^{(r)})|$$

But at the initial stage of the linearization procedure (see Equation (9)) the nonlinear residual $|f(\boldsymbol{x}^{(r)})|$ is usually relatively large and the search direction $\delta\boldsymbol{x}^{(r+1)}$ does not need to be very accurate. In this case a very accurate solution to the linear system (12) is not actually worth computing. Using a projection solver it is possible to adapt dynamically the exit tolerance ε_l without delaying the local convergence of the linearization procedure by using an inexact Newton approach [23]. Suitable choices of ε_l have been suggested by many authors [23–25]. Of course this would not be possible with a direct solver.

3.5. Other strategies for the preconditioner implementation

Being the linear system solution required several times within each nonlinear iteration, rather than recalculating the preconditioner any time a more efficient procedure might be to calculate a higher quality preconditioner only once at the first nonlinear iteration within each time step and keep it untouched during the successive iterations. This opportunity is experimented with using BiCGSTAB-SPLIT preconditioned with ILUT(ρ,τ). The threshold tolerance τ and the optimal fill-in value ρ are selected empirically (Table VII).

The timing from BiCGSTAB-SPLIT and ILUT(ρ,τ) is summarized in Table VIII where the fill-in influence is also shown. Interestingly observe that the best outcome is obtained with the smallest degree of fill-in $\rho = 5$. This appears to be the most appropriate trade-off

Table VII. Timing results after 300 s of TCE infiltration.

	BiCGSTAB-SPLIT	MA41-SPLIT
Total linear system solution time (s)	2653	1 133 140
Time per nonlinear iteration (s)	5.01	2138
Total # of time steps	67	67
Total # of back stepping	24	24
Total # of nonlinear iterations	530	530
Total # of linear iterations	14 531	—
Avg. linear iterations per time step	216	—
Avg. linear iterations per nonlinear iteration	27	—

Simulations performed on a 32-bit PC-workstation equipped with a 1526 MHz AMD processor, 2000 Mbyte of core memory, and 256 kbyte of secondary cache.
*$N \times N$ linear system.

Table VIII. Timing results after 300 s of TCE infiltration obtained with SPLIT and the linear solution performed with ILUT(ρ,τ) preconditioned BiCGSTAB with different fill-in values.

	ILUT $(7, 10^{-5})$	ILUT $(6, 10^{-5})$	ILUT $(5, 10^{-5})$
Total linear system solution time (s)	2252	2204	2341
Total simulation time (s)	2764	2697	2840
Total # of time steps	67	67	67
Total # of back stepping	24	24	24
Total # of nonlinear iterations	530	530	530
Total # of linear iterations	11 826	13 115	15 267
Avg. linear iterations per time step	176	196	227
Avg. linear iterations per nonlinear iteration	22	25	28

between the preconditioner quality and the computational effort required by its calculation. With $\rho = 5$ the computational cost relative to BiCGSTAB-ORD1 preconditioned with ILU(0) is reduced from 3143 to 2840 s (almost 10% faster). On balance the repeated calculation of the incomplete factor with ILU(0) appears to be substantially equivalent to the use of the better factor obtained with ILUT(ρ, τ) and computed only once at each nonlinear iteration.

4. CONCLUSIONS

The PDEs governing two phase flow through 3D porous media are integrated by the use of tetrahedral FE in space and Euler-backward FD in time. The resulting numerical equations are highly nonlinear. Nonlinearity is addressed via an iterative Picard-like solution scheme. Within each simple linearized iteration projection CG-like solvers such GMRES, TFQMR, and BiCGSTAB are used. Solver convergence is accelerated by preconditioning based on the incomplete factorization of the coefficient matrix with either partial or zero fill-in. The projection solvers are implemented with two nodal ordering ORD1 and ORD2 and a partitioned approach (SPLIT) wherein the unknown saturations are first eliminated and substituted into the pressure equations. The most efficient algorithm (BiCGSTAB-SPLIT) is compared with the direct solver (MA41-SPLIT) of the HSL. The main results are summarized below:

1. Ordering ORD2 yields a slightly better results than ordering ORD1 due to its smaller bandwidth which leads to a better preconditioner.
2. SPLIT turns out to be more cost effective than both ORD1 and ORD2 by a factor of almost 3. Moreover the core memory requirement is also markedly less.
3. BiCGSTAB is on balance superior to both TFQMR and GMRES(k) while not requiring the assessment of any empirical parameter.
4. Preconditioner ILU(0) appears to be quite appropriate. Although ILU(ρ, τ) may improve convergence, the resulting benefit is offset by the additional computational cost needed for its calculation.
5. The direct solver MA41-SPLIT is orders of magnitude more time-consuming than any of the iterative solvers experimented with in the present analysis.

APPENDIX A: SPATIAL DISCRETIZATION

PDEs (5) are discretized in space using linear tetrahedral FE and the Galerkin formulation. The FE solution to the coupled system and p_c are written as

$$p_w \approx \hat{p}_w = \sum_{l=1}^{N} \hat{p}_{w,l}(t) W_l(\mathbf{x})$$

$$S_w \approx \hat{S}_w = \sum_{l=1}^{N} \hat{S}_{w,l}(t) W_l(\mathbf{x}) \tag{A1}$$

$$p_c \approx \hat{p}_c = \sum_{l=1}^{N} \hat{p}_{c,l}(t) W_l(\mathbf{x})$$

where N is the number of FE nodes, W_l the basis (or coordinate) function associate to node l, and $\hat{p}_{\mathrm{w},l}$ and $\hat{S}_{\mathrm{w},l}$ the components of the nodal solution vectors $\boldsymbol{p}_{\mathrm{w}}$ and $\boldsymbol{S}_{\mathrm{w}}$, respectively.

A.1. Mass balance equation for the water phase

Substituting the approximate solution and the capillary pressure into the mass balance equation of the water phase the following residual is obtained:

$$L_1(\hat{p}_{\mathrm{w}}, \hat{S}_{\mathrm{w}}) = \frac{\partial(\phi \rho_{\mathrm{w}} \hat{S}_{\mathrm{w}})}{\partial t} - \nabla \cdot [\rho_{\mathrm{w}} \lambda_{\mathrm{w}} \underline{\underline{k}} (\nabla \hat{p}_{\mathrm{w}} - \rho_{\mathrm{w}} \mathbf{g})] - \mathbf{q}_{\mathrm{w}} \tag{A2}$$

Prescribing the orthogonally between the residual and the basis functions yields the Galerkin integral:

$$\int_{\Omega} L_1(\hat{p}_{\mathrm{w}}, \hat{S}_{\mathrm{w}}) W_j(\mathbf{x}) = 0, \quad j = 1, \ldots, N \tag{A3}$$

We assume that the coordinate directions are parallel to the principal directions of hydraulic anisotropy, so that the off-diagonal components of the conductivity tensor $\underline{\underline{k}}$ are zero:

$$\underline{\underline{k}} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$$

Expanding Equation (A3) and using Green's lemma for the spatial derivative term lead to:

$$\int_{\Omega} \frac{\partial(\phi \rho_{\mathrm{w}} \hat{S}_{\mathrm{w}})}{\partial t} W_j \, \mathrm{d}\Omega + \int_{\Omega} [\rho_{\mathrm{w}} \lambda_{\mathrm{w}} \underline{\underline{k}} \, (\nabla \hat{p}_{\mathrm{w}} - \rho_{\mathrm{w}} \mathbf{g})] \cdot \nabla W_j \, \mathrm{d}\Omega$$
$$- \int_{\Gamma} [\rho_{\mathrm{w}} \lambda_{\mathrm{w}} \underline{\underline{k}} (\nabla \hat{p}_{\mathrm{w}} - \rho_{\mathrm{w}} \mathbf{g})] \cdot \mathbf{n} W_j \, \mathrm{d}\Gamma - \int_{\Omega} \mathbf{q}_{\mathrm{w}} W_j \, \mathrm{d}\Omega = 0, \quad j = 1, \ldots, N$$

A.2. Mass balance equation for the non-wetting phase

Following a similar procedure for the non-wetting mass balance equation yields the expression:

$$\int_{\Omega} \frac{\partial(\phi \rho_{\mathrm{n}} (1 - \hat{S}_{\mathrm{w}}))}{\partial t} W_j \, \mathrm{d}\Omega + \int_{\Omega} [\rho_{\mathrm{n}} \lambda_{\mathrm{n}} \underline{\underline{k}} (\nabla \hat{p}_{\mathrm{w}} + \nabla \hat{p}_{\mathrm{c}} - \rho_{\mathrm{n}} \mathbf{g})] \cdot \nabla W_j \, \mathrm{d}\Omega$$
$$- \int_{\Gamma} [\rho_{\mathrm{n}} \lambda_{\mathrm{n}} \underline{\underline{k}} (\nabla \hat{p}_{\mathrm{w}} + \nabla \hat{p}_{\mathrm{c}} - \rho_{\mathrm{n}} \mathbf{g})] \cdot \mathbf{n} W_j \, \mathrm{d}\Gamma - \int_{\Omega} \mathbf{q}_{\mathrm{n}} W_j \, \mathrm{d}\Omega = 0, \quad j = 1, \ldots, N$$

A.3. Stiffness and mass matrix coefficients

Substituting Equation (A1) and making use of boundary conditions to replace the above boundary integral terms lead to the following system of ordinary differential equations:

$$H(\boldsymbol{x})\boldsymbol{x} + M(\boldsymbol{x})\dot{\boldsymbol{x}} + \boldsymbol{q}(\boldsymbol{x}) = 0 \tag{A4}$$

with matrices H and M given by

$$H = \begin{bmatrix} H_{\mathrm{w}} & 0 \\ H_{\mathrm{n}} & 0 \end{bmatrix}, \quad M = \begin{bmatrix} 0 & M_{\mathrm{w}} \\ 0 & M_{\mathrm{n}} \end{bmatrix}$$

while vectors $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, and $\boldsymbol{q}$ read:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{\mathrm{w}} \\ \boldsymbol{S}_{\mathrm{w}} \end{bmatrix}, \quad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}}_{\mathrm{w}} \\ \dot{\boldsymbol{S}}_{\mathrm{w}} \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{w}} \\ \boldsymbol{q}_{\mathrm{n}} \end{bmatrix}$$

where the coefficients have the following expressions:

$$h_{\mathrm{w},ij} = \sum_{e=1}^{E} \int_{V^e} \rho_{\mathrm{w}}^e \lambda_{\mathrm{w}}^e \underline{\underline{k}}^e \nabla W_i^e \cdot \nabla W_j^e \, \mathrm{d}V, \quad h_{\mathrm{n},ij} = \sum_{e=1}^{E} \int_{V^e} \rho_{\mathrm{n}}^e \lambda_{\mathrm{n}}^e \underline{\underline{k}}^e \nabla W_i^e \cdot \nabla W_j^e \, \mathrm{d}V$$

$$m_{\mathrm{w},ij} = \sum_{e=1}^{E} \int_{V^e} \phi^e \rho_{\mathrm{w}}^e W_i^e W_j^e \, \mathrm{d}V, \quad m_{\mathrm{n},ij} = \sum_{e=1}^{E} \int_{V^e} \phi^e \rho_{\mathrm{n}}^e W_i^e W_j^e \, \mathrm{d}V$$

$$q_{\mathrm{w},i} = -\sum_{e=1}^{E} \left[\int_{V^e} g(\rho_{\mathrm{w}}^e)^2 \lambda_{\mathrm{w}}^e k_z^e \frac{\partial W_i^e}{\partial z} \, \mathrm{d}V + \int_{V^e} q_{\mathrm{w}}^e W_i^e \, \mathrm{d}V^e + \int_{\Gamma_2^e} q_{\mathrm{w}}^* W_i^e \, \mathrm{d}\Gamma \right]$$

$$q_{\mathrm{n},i} = \sum_{e=1}^{E} \left[\left(\int_{V^e} \rho_{\mathrm{n}}^e \lambda_{\mathrm{n}}^e \underline{\underline{k}}^e \nabla W_i^e \cdot \nabla W_j^e \, \mathrm{d}V \right) \hat{p}_{\mathrm{c},j} \right.$$

$$\left. - \int_{V^e} g(\rho_{\mathrm{n}}^e)^2 \lambda_{\mathrm{n}}^e k_z^e \frac{\partial W_i^e}{\partial z} \, \mathrm{d}V - \int_{V^e} q_{\mathrm{n}}^e W_i^e \, \mathrm{d}V - \int_{\Gamma_2^e} q_{\mathrm{n}}^* W_i^e \, \mathrm{d}\Gamma \right]$$

In the above equations, $H_{\mathrm{w}} = \{h_{\mathrm{w},ij}\}$, $H_{\mathrm{n}} = \{h_{\mathrm{n},ij}\}$, $M_{\mathrm{w}} = \{m_{\mathrm{w},ij}\}$, and $M_{\mathrm{n}} = \{m_{\mathrm{n},ij}\}$ represent water and non-wetting stiffness and mass matrices, respectively; $\boldsymbol{q}_{\mathrm{w}} = \{q_{\mathrm{w},i}\}$ and $\boldsymbol{q}_{\mathrm{n}} = \{q_{\mathrm{n},i}\}$ incorporate gravity terms, source/sinks terms, and Neumann boundary conditions. Vectors $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ contain the unknown nodal water pressure and saturation values, and the corresponding time derivatives. Mass matrices M_{w} and M_{n} are lumped for stability reasons, while in the stiffness matrices H_{w} and H_{n} the hydraulic mobility can be evaluated 'fully upwind', with the fluid potential of each phase α defined as

$$\psi_\alpha = p_\alpha - \rho_\alpha g z, \quad \alpha = \mathrm{w}, \mathrm{n}$$

and the upwind mobility along each side connecting two nodal points evaluated as

$$\lambda_{\alpha,ij} = \lambda_{\alpha,i}, \quad \psi_{\alpha,j} - \psi_{\alpha,i} \leqslant 0$$

$$\lambda_{\alpha,ij} = \lambda_{\alpha,j}, \quad \psi_{\alpha,j} - \psi_{\alpha,i} > 0$$

This choice introduces a small amount of numerical diffusion which guarantees a monotonic behaviour of the numerical solution and avoids undesirable numerical oscillations.

ACKNOWLEDGEMENTS

This study has been partially funded by the Italian PRIN-MIUR project 'Numerical Models for Multiphase Flow and Deformation in Porous Media'.

REFERENCES

1. Aziz K, Settari A. *Petroleum Reservoir Simulation*. Applied Science: London, 1979.
2. Helmig R. *Multiphase Flow and Transport Processes in the Subsurface. A Contribution to the Modeling of Hydrosystem*. Springer: Berlin, 1997.
3. Brooks RH, Corey AT. Hydraulic properties of porous media. *Hydrology Paper 3*, Colorado State University, Fort Collins, CO, 1964.
4. Kaluarachchi JJ, Parker JC. An efficient finite element method for modeling multiphase flow. *Water Resources Research* 1989; **25**(1):43–54.
5. Peaceman DW. *Fundamentals of Numerical Reservoir Simulation*. Elsevier: Amsterdam, 1977.
6. Helmig R, Huber R. Comparison of Galerkin-type discretization techniques for two phase flow in heterogeneous porous media. *Advances in Water Resources* 1998; **21**(8):697–711.
7. Huber R, Helmig R. Node-centered finite volume discretizations for the numerical simulation of multiphase flow in heterogeneous porous media. *Computational Geosciences* 2000; **4**:141–164.
8. Dennis JE, Schnabel RB. *Numerical Methods for Unconstrained Optimization*. Prentice-Hall: Englewood Cliffs, NJ, 1983.
9. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. SIAM: Philadelphia, PA, 1995.
10. Paniconi C, Putti M. A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems. *Water Resources Research* 1994; **30**(12):3357–3374.
11. Putti M, Paniconi C. Picard and Newton linearization for the coupled model of saltwater intrusion in aquifers. *Advances in Water Resources* 1995; **18**(3):159–170.
12. Celia MA, Boutoulas ET, Zarba RL. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research* 1990; **26**(7):1483–1496.
13. Paniconi C, Aldama AA, Wood EF. Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation. *Water Resources Research* 1991; **27**(6):1147–1163.
14. Saad Y, Schultz MH. GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**(3):856–869.
15. van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:631–644.
16. Freund RW. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM Journal on Scientific Computing* 1993; **14**:470–482.
17. Saad Y. *Iterative Methods for Sparse Linear Systems*. SIAM: Philadelphia, PA, 2003.
18. Kershaw DS. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics* 1978; **26**:43–65.
19. Sauter S. The ILU method for finite-element discretizations. *Journal of Computational and Applied Mathematics* 1991; **36**(1):91–106.
20. Saad Y. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications* 1994; **1**:387–402.
21. Comerlati A, Pini G, Gambolati G. Linearized solutions to two phase flow in porous media solved by finite elements. In *ICNAAM—International Conference of Numerical Analysis and Applied Mathematics*, Simos TE, Tsitouras C (eds). Wiley-VCH: New York, 2004; 106–109.

22. HSL Archive—a catalogue of subroutines. Aea Technology, Engineering Software, CCLRC. http://www.cse.clrc.ac.uk/nag/hsl, 2004.
23. Eisenstat SC, Walker HF. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing* 1996; **17**(1):16–32.
24. Brown PN, Saad Y. Convergence theory of nonlinear Newton–Krylov algorithms. *SIAM Journal on Scientific Optimization* 1994; **4**(2):297–330.
25. Bastian P, Helmig R. Efficient fully-coupled solution techniques for two-phase flow in porous media. Parallel multigrid solution and large scale computations. *Advances in Water Resources* 1999; **23**:199–216.

