Advances in Engineering Software 46 (2012) 63–68

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Quasi-Newton acceleration of ILU preconditioners for nonlinear two-phase flow equations in porous media

L. Bergamaschi [a], R. Bru [b,*], A. Martínez [a], M. Putti [a]

[a] Department of Mathematical Methods and Models for Scientific Applications, University of Padova, Italy
[b] Instituto de Matemática Multidisciplinar, Departamento de Matemática Aplicada, Universidad Politecnica de Valencia, Spain

ARTICLE INFO

Article history:
Available online 4 December 2010

Keywords:
Quasi-Newton method
Krylov iterations
Updating preconditioners
Inexact Newton method
Two-phase flow
Porous media

ABSTRACT

In this work, preconditioners for the iterative solution by Krylov methods of the linear systems arising at each Newton iteration are studied. The preconditioner is defined by means of a Broyden-type rank-one update of a given initial preconditioner, at each nonlinear iteration, as described in [5] where convergence properties of the scheme are theoretically proved. This acceleration is employed in the solution of the nonlinear system of algebraic equations arising from the finite element discretization of two-phase flow model in porous media. We report numerical results of the application of this approach when the initial preconditioner is chosen to be the incomplete LU decomposition of the Jacobian matrix at the initial nonlinear stage. It is shown that the proposed acceleration reduces the number of linear iterations needed to achieve convergence. Also, the cost of computing the preconditioner is reduced as this operation is made only once at the beginning of the Newton iteration.

1. Introduction

Newton's method requires the solution of a number of linear systems with the Jacobian J as the coefficient matrix. When J is large and sparse, e.g. for problems arising from the discretization of a nonlinear PDE, preconditioned Krylov based iterative schemes can be employed for the solution of the linear system. As a result, two nested iterative procedures need to be implemented. To avoid oversolving, i.e. excessive and essentially useless iterations of the inner scheme, it is crucial to employ an "inexact" technique [10]. This approach tries to control the number of linear iterations by allowing the accuracy of the linear solver to vary across nonlinear iterations [12].

There are many papers in the literature, trying to combine the properties of the two nested iterative procedures. Among these we quote [8,21] where multilevel preconditioners are proposed for accelerating the solution of the Jacobian linear system. In [20] the authors propose to exploit the underlying Krylov subspace information from the linear solver in order to accelerate the nonlinear convergence of Newton–Krylov methods. In [9] the authors try to accelerate the Newton convergence by making use of the information gathered from the Krylov subspace generated by the GMRES linear solver. Differently from their approach, we study how preconditioning efficiency can be enhanced as the nonlinear iteration progresses. In fact, another crucial issue for the reduction of total linear iterations is to use efficient preconditioning techniques. In general, ILU-type preconditioners [22,23] can be employed and calculated at every nonlinear iteration. Techniques for selectively evaluating a preconditioner P may be developed to save on the cost of the calculation of the preconditioner. Note that the two phases where efficiency can be mostly improved are the cost of the linear system solution (thus including the number of iterations) and the cost of preconditioner evaluation.

In this paper we are mainly concerned with the efficient preconditioning of the linear system. The "optimal" preconditioner P is aimed at clustering eigenvalues of $PJ(\mathbf{x}_k)$. This can be accomplished for instance by minimizing the constant C of:

$$\|zI - PJ(\mathbf{x}_k)\| \leqslant C, \qquad z \in \mathscr{R}. \tag{1}$$

This requires that information from the nonlinear iterative scheme be taken into account in the evaluation of P.

The approach proposed in this paper is to solve the inner systems of the Newton method with an iterative Krylov subspace method, starting with ILU(0) [22] computed from the initial Jacobian and to update this preconditioner using a rank one sum. A sequence of preconditioners P_k can thus be defined by imposing the secant condition, as used in the implementation of Quasi-Newton methods [11]. We choose to work with the Broyden update as described for instance in [19], and analyze the theoretical properties of the preconditioner and the numerical behavior of the resulting scheme. We are aware that the choice $P_0 \equiv$ ILU(0) is not the only possible alternative. Among the others we mention the class of

* Corresponding author.
E-mail addresses: berga@dmsa.unipd.it (L. Bergamaschi), rbru@mat.upv.es (R. Bru), acalomar@dmsa.unipd.it (A. Martínez), putti@dmsa.unipd.it (M. Putti).

0965-9978/$ - see front matter © 2010 Civil-Comp Ltd and Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2010.10.011

approximate inverse preconditioners (see [6,2,3,14–16]) which are particularly suited in a parallel environment. We choose to work with the incomplete ILU factorization because of its wide-spread use in this type of problems and its simplicity. Our strategy is aimed to be independent of the initial preconditioner choice, i.e. the scope of this paper is to construct a sequence of preconditioners that improves the initial one.

We successfully try our approach on a number of nonlinear problems of large size arising from the finite element (FE) discretization of two phase flow equations in porous media. The Broyden acceleration provides in all the test cases a reduction of roughly 15% of the number of linear iterations and 10% of the total CPU time.

The paper is organized as follows. The equations governing the two phase flow model are given in Section 2. In Section 3, the numerical treatment of the governing equations and a general approach of the obtained nonlinear system are studied. The Broyden-type rank-one update of the initial preconditioner is discussed in Section 4, where the algorithm is described. In Section 5, we report numerical results in 2D and 3D cases. Finally, some conclusions are given in Section 6.

2. Two phase-flow model: governing equations

Immiscible two-phase flow in porous media in isothermal conditions is described by the mass conservation equation [1,17]:

$$\frac{\partial(\phi\ \rho_\alpha\ S_\alpha)}{\partial t} = -\nabla\cdot[\rho_\alpha v_\alpha] + q_\alpha \qquad \alpha \in w,n, \tag{2}$$

where subscript α refers to wetting (w) and non-wetting (n) phase, respectively (e.g. water and oil or water and gas). For each phase, S_α is the saturation, ρ_α the density, v_α the Darcy velocity, and q_α the mass source/sink rate. Finally, ϕ denotes the porous medium porosity. The phase velocity is given by extending Darcy law:

$$v_\alpha = -\lambda_\alpha\ \underline{\underline{k}}\ (\nabla p_\alpha - \rho_\alpha\ g), \tag{3}$$

where the mobility λ_α is defined as the ratio between relative permeability $k_{r\alpha}$ and dynamic viscosity μ_α, $\underline{\underline{k}}$ is the intrinsic permeability tensor, p_α the α-phase pressure, and g the gravity acceleration vector. Substitution of Eq. (3) into the continuity Eq. (2) yields:

$$\frac{\partial(\phi\ \rho_\alpha\ S_\alpha)}{\partial t} = \nabla\cdot[\rho_\alpha\lambda_\alpha\ \underline{\underline{k}}(\nabla p_\alpha - \rho_\alpha\ g)] + q_\alpha. \tag{4}$$

The solution of the PDE system (4) requires the following auxiliary relationships:

$$S_w + S_n = 1; \qquad p_c(S_w) = p_n - p_w, \tag{5}$$

where p_c, the capillary pressure, is defined as the difference between the non-wetting and the wetting phase pressures. Substituting the auxiliary relationships (5) into the PDEs (4), and solving for p_w and S_w, the system can be written as:

$$\begin{aligned}\frac{\partial(\phi\ \rho_w\ S_w)}{\partial t} &= \nabla\cdot[\rho_w\lambda_w\ \underline{\underline{k}}(\nabla p_w - \rho_w\ g)] + q_w\\ \frac{\partial[\phi\ \rho_n\ (1-S_w)]}{\partial t} &= \nabla\cdot[\rho_n\lambda_n\ \underline{\underline{k}}(\nabla p_w + \nabla p_c - \rho_n\ g)] + q_n.\end{aligned} \tag{6}$$

Appropriate initial and boundary conditions complete the model formulation.

Capillary properties can be described using a number of constitutive laws, whose most widely used models are Brooks–Corey (BC) [7] and Van Genuchten (VG) [25]. Brooks–Corey capillary laws have the following representation:

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

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. Van Genuchten constitutive laws read as:

$$k_{rw}(S_w) = S_{we}^{1/2}\ [1-(1-S_{we}^{1/m})^m]^2; \qquad k_{rn}(S_w) = (1-S_e)^{1/2}\ (1-S_{we}^{1/m})^{2m}$$
$$p_c(S_w) = p_0\ (S_{we}^{-1/m} - 1)^{1-m},$$

where p_0 is the characteristic capillary pressure of the medium, and m is related to the pore distribution. Eq. (6) represent a highly nonlinear system of PDEs and fluid densities and viscosities may also depend on the corresponding phase pressure:

$$\rho_w = \rho_w(p_w); \qquad \rho_n = \rho_n(p_n); \qquad \mu_w = \mu_w(p_w); \qquad \mu_n = \mu_n(p_n).$$

3. Numerical model

3.1. Two-phase flow finite element equations

Eq. (6) are discretized in space using linear finite elements (triangles in 2D and tetrahedra in 3D) yielding a system of first order differential equations that reads:

$$\begin{bmatrix} H_w & M_w \\ H_n & M_n \end{bmatrix}\begin{bmatrix} \boldsymbol{p}_w \\ \boldsymbol{S}_w \end{bmatrix} + \begin{bmatrix} 0 & M_w \\ 0 & M_n \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{p}}_w \\ \dot{\boldsymbol{S}}_w \end{bmatrix} + \begin{bmatrix} \boldsymbol{q}_w \\ \boldsymbol{q}_n \end{bmatrix} = 0, \tag{7}$$

where H_w, H_n, M_w and M_n are wetting and non-wetting stiffness and mass matrices; $[\boldsymbol{q}_w, \boldsymbol{q}_w]^T$ incorporate source/sink 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, while in the stiffness matrices H_w and H_n hydraulic mobility is evaluated "fully upwind" [1,17,18] 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 and positive definite and symmetric and positive semi-definite, respectively. Mass matrices M_w and M_n are diagonal matrices. System (7) can be written in a more compact form as:

$$H\ \boldsymbol{x} + M\ \dot{\boldsymbol{x}} + \boldsymbol{q} = 0, \tag{8}$$

where the meaning of the new symbols is derived by comparison of Eqs. (7) and (8). The time integration is implemented via Euler backward FD, giving the following nonlinear system of algebraic equations:

$$\left[H + \frac{M}{\Delta t}\right]^{(m+1)} \boldsymbol{x}^{(m+1)} = \left[\frac{M}{\Delta t}\right]^{(m+1)} \boldsymbol{x}^{(m)} - \boldsymbol{q}^{(m+1)}, \tag{9}$$

where Δt is the time step size; (m) and $(m+1)$ indicate the previous and the current time level, respectively.

3.2. General approach for nonlinear systems

The nonlinear system (9), which has the same form for both two phase and unsaturated flow, is solved by Newton-like iterative methods. To this aim, Eq. (9) is rewritten as:

$$F(\boldsymbol{x}^{(m+1)}) = A\ \boldsymbol{x}^{(m+1)} - \bar{\boldsymbol{q}} = 0, \tag{10}$$