Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 196 (2007) 2647–2656
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Novel preconditioners for the iterative solution to FE-discretized coupled consolidation equations

Luca Bergamaschi, Massimiliano Ferronato *, Giuseppe Gambolati

Department of Mathematical Methods and Models for Scientific Applications, University of Padova, via Trieste 63, 35121 Padova, Italy

Received 14 November 2006; received in revised form 24 January 2007; accepted 26 January 2007

Abstract

A major computational issue in the finite element (FE) integration of coupled consolidation equations is the repeated solution in time of the resulting discretized indefinite system. Because of ill-conditioning, the iterative solution, which is recommended in large size 3D settings, requires the computation of a suitable preconditioner to guarantee convergence. In this paper the coupled system is solved by a Krylov subspace method preconditioned by an inexact constraint preconditioner (ICP) preserving the same block structure as the native FE matrix. The conditioning number of the preconditioned coupled problem depends on the quality of the approximation of the block corresponding to the structural stiffness matrix. An efficient algorithm to implement ICP into a Krylov subspace method is developed. Numerical tests performed on realistic 3D problems reveal that ICP typically outperforms standard ILUT preconditioners and proves much more robust in severely ill-conditioned problems.

Keywords: Preconditioning; Krylov subspace methods; Coupled consolidation

1. Introduction

The time-dependent distribution of displacements and fluid pressure in porous media is governed by the consolidation theory. This was first mathematically described by Biot [1], who coupled the elastic equilibrium equations with a continuity or mass balance equation to be solved under appropriate boundary and initial flow and loading conditions.

The coupled consolidation equations are typically solved numerically using finite elements (FE) in space, thus giving rise to a system of first-order differential equations the solution to which can be addressed by an appropriate time marching scheme. A major computational issue is the repeated solution in time of the resulting discretized indefinite equations. In particular, with the small time integration steps typically required in the early phase of the analysis the final linear system may be severely ill-conditioned [2], so that obtaining an accurate solution may prove difficult with any numerical approach.

Because of the large size of realistic three-dimensional (3D) consolidation models (and particularly so in problems related to fluid withdrawal/injection from/into geological formations) the use of iterative solvers is strongly recommended. Among them, projection (or conjugate gradient-like) methods based on Krylov subspaces for unsymmetric indefinite systems, such as Bi-CGSTAB (bi-conjugate gradient stabilized [3]), are attracting a growing interest on the grounds of their robustness and efficiency [4–9]. However, a key issue to guarantee and accelerate convergence is the selection of an ad hoc efficient preconditioning strategy, which must prove both a good and inexpensive approximation of the inverse of the coupled native matrix. The ILUT preconditioner [10], based on an incomplete triangular factorization with controlled fill-in and supplemented with a preliminary left and right scaling [6], has proved one of the most robust and efficient tools for the

* Corresponding author. Tel.: +39 049 8271332; fax: +39 049 8271333.
E-mail address: ferronat@dmsa.unipd.it (M. Ferronato).
URL: http://www.dmsa.unipd.it/~ferronat (M. Ferronato).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.01.013

iterative solution to the FE coupled consolidation equations. In particular, the use of a proper preliminary scaling, such as the Least Square Log (LSL) algorithm [7], helps stabilize CG-like methods to round-off errors with a significant increase of ILUT robustness in severely ill-conditioned problems.

The block matrix structure arising from the FE discretization of generally coupled problems may suggest the use of block preconditioners. For example, a recent work [11] discusses the development of a block ILU preconditioner to solve a strongly coupled system discretizing a fluid–structure interaction. The coefficient matrix of FE consolidation equations can be also viewed as an example of saddle point problem as it typically arises in the discretization of Navier–Stokes equations, where diagonal block preconditioners have been successfully employed [12]. To accelerate Krylov solvers in the solution of saddle point problems the so-called "Constraint Preconditioners" have been first introduced in constrained optimization [13]. This terminology has been preserved in other fields as well, including least squares and also Navier–Stokes equations [13–18]. For a thorough review of the constraint preconditioning see also [19] and references therein. The aim of the present paper is to investigate the performance and the robustness of a novel Inexact Constraint Preconditioner (ICP) developed for the solution to the symmetric indefinite system arising from the FE integration of the coupled consolidation equations. After a brief review of the FE coupled consolidation equations, the ICP properties are summarized and an efficient algorithm is proposed for the implementation into a Krylov subspace method. The ICP performance with two realistic medium and large size 3D problems is compared to that of the Exact Constraint Preconditioner and the LSL-ILUT strategy with optimal fill-in degree. In particular, the comparison is performed for both normally conditioned and severely ill-conditioned problems. Finally, the ICP potential for real long-term simulations is addressed and some conclusive remarks close the paper.

2. Finite element coupled consolidation equations

The system of partial differential equations governing the 3D coupled consolidation process in fully saturated porous media is based on the classical Biot's formulation [1] as modified by van der Knaap [20] and Geertsma [21]:

$$(\lambda+\mu)\frac{\partial \epsilon}{\partial i}+\mu\nabla^2 u_i=\alpha\frac{\partial p}{\partial i}\quad i=x,y,z, \tag{1}$$

$$\frac{1}{\gamma}\nabla(k\nabla p)=[\phi\beta+c_{\mathrm{br}}(\alpha-\phi)]\frac{\partial p}{\partial t}+\alpha\frac{\partial \epsilon}{\partial t}, \tag{2}$$

where c_{br} and β are the volumetric compressibility of solid grains and water, respectively, ϕ is the porosity, k the medium hydraulic conductivity, α the Biot coefficient, λ and μ are the Lamé constant and the shear modulus of the porous medium, respectively, γ is the specific weight of water, ∇ the gradient operator, x, y, z are the coordinate directions, and t is time. The independent variables are the incremental pore pressure p and the components of incremental displacement u_i along the i-direction, with the medium volumetric dilatation ϵ equal to $\sum_i \partial u_i/\partial i$.

Integration by FE in space yields a system of first-order differential equations which can be written as

$$\begin{bmatrix} K & -Q \\ 0 & H \end{bmatrix}\begin{Bmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{Bmatrix}+\begin{bmatrix} 0 & 0 \\ Q^{\mathrm{T}} & P \end{bmatrix}\begin{Bmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{Bmatrix}=\begin{Bmatrix} \boldsymbol{f}^u \\ \boldsymbol{f}^p \end{Bmatrix}, \tag{3}$$

where K, H, P and Q are the elastic stiffness, flow stiffness, flow capacity and flow-stress coupling matrices, respectively, $\{\boldsymbol{u},\boldsymbol{p}\}^{\mathrm{T}}$ and $\{\dot{\boldsymbol{u}},\dot{\boldsymbol{p}}\}^{\mathrm{T}}$ are the vectors of the unknown variables u_i and p and the corresponding time derivatives, and $\{\boldsymbol{f}^u,\boldsymbol{f}^p\}^{\mathrm{T}}$ is the vector of the nodal loads ($\boldsymbol{f}^u$) and flow sources ($\boldsymbol{f}^p$).

Eq. (3) can be written in a more compact form as

$$K_1\boldsymbol{x}+K_2\dot{\boldsymbol{x}}+\boldsymbol{f}=0, \tag{4}$$

where the meaning of the new symbols above is immediately derived from comparison of Eqs. (3) and (4). Eq. (4) is integrated in time by the well known θ-method (e.g. [22])

$$\left[\theta K_1+\frac{K_2}{\Delta t}\right]\boldsymbol{x}_{m+1}=\left[\frac{K_2}{\Delta t}-(1-\theta)K_1\right]\boldsymbol{x}_m - [\theta\boldsymbol{f}_{m+1}+(1-\theta)\boldsymbol{f}_m], \tag{5}$$

where Δt is the time integration step.

Eq. (5) is to be repeatedly solved to obtain the displacement and the pore pressure in time. The non-symmetric matrix controlling the solution scheme reads

$$A=\left[\theta K_1+\frac{K_2}{\Delta t}\right]=\begin{bmatrix} \theta K & -\theta Q \\ \frac{Q^{\mathrm{T}}}{\Delta t} & \theta H+\frac{P}{\Delta t} \end{bmatrix}. \tag{6}$$

Matrix A can be readily symmetrized by multiplying the upper set of equations by $1/\theta$ and the lower set by $-\Delta t$, thus obtaining the following sparse 2×2 block symmetric indefinite matrix:

$$\mathscr{A}=\begin{bmatrix} K & B^{\mathrm{T}} \\ B & -C \end{bmatrix}, \tag{7}$$

where $B=-Q^{\mathrm{T}}$ and $C=\theta\Delta t H+P$. The blocks K and C are both symmetric and positive definite (SPD). In 3D problems, denoting by n the number of FE nodes, $C\in\Re^{n\times n}$, $B\in\Re^{n\times 3n}$, and $K\in\Re^{3n\times 3n}$.

The set of Eq. (5) is unconditionally stable for any Δt provided that $\theta\geqslant 0.5$ [22]. If $\theta<0.5$ the following upper bound for the time step holds:

$$\Delta t<\frac{2}{1-2\theta}\frac{1}{\vartheta_1}, \tag{8}$$

where ϑ_1 is the largest eigenvalue of the generalized SPD eigenproblem [5]

$$H\boldsymbol{v}=\vartheta(Q^{\mathrm{T}}K^{-1}Q+P)\boldsymbol{v}. \tag{9}$$

The main difficulty with the repeated solution to (5), however, is the possible ill-conditioning of matrices (6) or (7) which depends on the interrelation between Δt, the hy-

dro-mechanical properties of the porous medium and the FE discretization. Ferronato et al. [2] have shown that a critical time step Δt_{crit} exists that can be defined as

$$\Delta t_{\text{crit}} = \chi(\psi,\theta)\frac{V\gamma}{kE}, \tag{10}$$

where E is the Young modulus of the porous medium, V a characteristic size of the FE grid (e.g. the elemental volume), $\psi = \phi\beta E$, and χ is a generally unknown dimensionless factor depending on ψ, θ, and the element distortion. For $\Delta t \leqslant \Delta t_{\text{crit}}$ the conditioning of A or $\mathscr{A}$ suddenly worsens with the solution to system (5) quite difficult to get independently of the selected solver. In long-term simulations a small Δt is typically needed in the early stage of the consolidation process, while larger values may be used as the system approaches the steady state. Hence, the initial steps are the most difficult and expensive ones, with the convergence generally accelerating as the simulation proceeds and Δt grows.

3. Constraint preconditioner

To solve the indefinite system

$$\mathscr{A}\boldsymbol{x} = \boldsymbol{b} \tag{11}$$

we employ a Krylov method preconditioned with $\mathscr{M}^{-1}$, where

$$\mathscr{M} = \begin{bmatrix} G & B^{\mathrm{T}} \\ B & -C \end{bmatrix} \tag{12}$$

and G is an SPD approximation of the structural stiffness matrix K. Its inverse G^{-1} can be viewed as a preconditioner for K, and it is assumed to be explicitly known.

The proposed preconditioner is intended to produce a cluster of eigenvalues of the iteration matrix $\mathscr{M}^{-1}\mathscr{A}$ around unity. In particular, it can be proved that, whatever G, there are at least n unit eigenvalues with the remainder bounded by the extreme eigenvalues of $G^{-1}K$ [23]. The success of $\mathscr{M}^{-1}$ in accelerating the convergence rests therefore on the following conditions:

(1) G^{-1} must be a good preconditioner of block K;
(2) the application of $\mathscr{M}^{-1}$ in the solver algorithm, i.e. solution of the linear system $\mathscr{M}\boldsymbol{y} = \boldsymbol{r}$ with $\boldsymbol{r}$ the residual vector, must be computationally as inexpensive as possible.

After some calculations, the (right preconditioned) iteration matrix can be written as

$$\mathscr{A}\mathscr{M}^{-1} = \begin{bmatrix} X & Y \\ 0 & I_n \end{bmatrix}, \tag{13}$$

where X is equal to $(K + B^{\mathrm{T}}C^{-1}B)(G + B^{\mathrm{T}}C^{-1}B)^{-1}$, Y is a $3n \times n$ block whose explicit expression is not essential for the discussion that follows, and I_n denotes the n-order identity matrix. It is well known that the classical Preconditioned Conjugate Gradient (PCG) method can be used with SPD matrices only and produces a sequence of residuals $\boldsymbol{r}_k = \mathscr{P}_k(\mathscr{A}\mathscr{M}^{-1})\boldsymbol{r}_0$, where $\mathscr{P}_k$ denotes a polynomial of degree k [24], k the iteration count and $\boldsymbol{r}_0$ the initial residual. Due to the block structure of the preconditioned matrix, if $\boldsymbol{r}_0 = [\hat{\boldsymbol{r}}_0, 0]^{\mathrm{T}}$, then

$$\boldsymbol{r}_k = \begin{bmatrix} \mathscr{P}_k(X)\hat{\boldsymbol{r}}_0 \\ 0 \end{bmatrix}. \tag{14}$$

Eq. (14) implies that the Conjugate Gradient method used with the preconditioned matrix (13) exhibits the same behavior as the Conjugate Gradient used with the block X, i.e. the SPD matrix $(K + B^{\mathrm{T}}C^{-1}B)$ preconditioned with the SPD matrix $(G + B^{\mathrm{T}}C^{-1}B)^{-1}$. Therefore, it may be concluded that the classical PCG algorithm can be successfully used with our indefinite linear system, as is also proved in Ref. [25], provided that the last n components of $\boldsymbol{r}_0$ are zero. To fulfil this requirement the iterative procedure is started with $\boldsymbol{x}_0 = \mathscr{M}^{-1}\boldsymbol{b}$

$$\boldsymbol{r}_0 = \boldsymbol{b} - \mathscr{A}\mathscr{M}^{-1}\boldsymbol{b} = \begin{bmatrix} I_{3n} - X & -Y \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{r}}_0 \\ 0 \end{bmatrix}. \tag{15}$$

The application of $\mathscr{M}^{-1}$ in the PCG algorithm requires at each iteration the computation of $\mathscr{M}^{-1}\boldsymbol{r} = \boldsymbol{y}$, i.e. the solution to the system $\mathscr{M}\boldsymbol{y} = \boldsymbol{r}$ (see Eq. (12)):

$$\begin{bmatrix} G & B^{\mathrm{T}} \\ B & -C \end{bmatrix}\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{bmatrix}. \tag{16}$$

A way to solve (16) relies on deriving $\boldsymbol{y}_1$ from the upper set of equations

$$\boldsymbol{y}_1 = G^{-1}(\boldsymbol{r}_1 - B^{\mathrm{T}}\boldsymbol{y}_2) \tag{17}$$

and substituting in the lower set, thus obtaining

$$S\boldsymbol{y}_2 = BG^{-1}\boldsymbol{r}_1 - \boldsymbol{r}_2 \tag{18}$$

with

$$S = (BG^{-1}B^{\mathrm{T}} + C). \tag{19}$$

Matrix S is the Schur complement of system (16) and can be computed explicitly provided that G^{-1} is known. Hence, the cost for applying $\mathscr{M}^{-1}$ basically rests on the efficient solution to the $n \times n$ linear SPD system (18) at each iteration. This can be done, for example, by an inner PCG iteration. The preconditioner $\mathscr{M}^{-1}$ will be referred to as Exact Constraint Preconditioner (ECP).

Unfortunately, solving system (18) at each iteration can be quite expensive also with a very sparse G^{-1}, thus affecting significantly the performance of the whole algorithm. To make the $\mathscr{M}^{-1}$ application cheaper, we can compute an approximate solution to (18) by the use of an approximation of S^{-1}. Using an incomplete Cholesky factorization of S, with either partial fill-in and drop tolerance τ_{I} (ILLT) or zero fill-in (IC(0))

$$S^{-1} \simeq \left(\tilde{L}\tilde{L}^{\mathrm{T}}\right)^{-1} \tag{20}$$

the true solution to system (18) is replaced by "inexpensive" forward and backward substitutions:

$$\left(\tilde{L}\tilde{L}^{\mathrm{T}}\right)\boldsymbol{y}_2 = BG^{-1}\boldsymbol{r}_1 - \boldsymbol{r}_2. \tag{21}$$

This is equivalent to applying a new preconditioner $\widehat{\mathscr{M}}^{-1}$ that no longer satisfies the ECP theoretical properties, with the convergence of PCG for our indefinite system no more guaranteed. Therefore preconditioner $\widehat{\mathscr{M}}^{-1}$ must be implemented into a Krylov method designed for generally unsymmetric and indefinite systems, such as for instance Bi-CGSTAB, and will be referred to as Inexact Constraint Preconditioner (ICP).

3.1. Choice of matrix G^{-1}

G^{-1} is a key factor for the overall quality of the proposed preconditioner. This matrix, which is to be explicitly known as is needed for the computation of S, should be a good preconditioner for the elastic stiffness block K.

The simplest and cheapest choice for G is diag(K), i.e. the Jacobi preconditioner. The related ICP has been successfully experimented with in a different context, i.e. interior point methods for quadratic optimization [26,16,27]. Unfortunately, as the elastic stiffness block is not diagonally dominant, the Jacobi preconditioner may prove so a poor approximation of K^{-1} as to preclude in many instances the solver convergence.

Recently a new class of preconditioners based on the sparse approximate inverses involving only matrix–vector products has been developed, see among others SPAI [28], AINV [29,30] and FSAI [31]. For an extensive comparative study of these preconditioners the reader is referred to Benzi and Tůma [32].

We elected to use AINV which is generally more efficient than FSAI and SPAI as accelerator of Krylov solvers on scalar machines, due to its flexibility in the generation of the pattern of the approximate inverse factor. Its cost and fill-in are controlled by a user-specified parameter τ_{A} equal to the fraction of the diagonal term below which an AINV coefficient is dropped. The larger τ_{A}, the cheaper and sparser AINV. The AINV preconditioner is provided in the convenient factorized form

$$K^{-1} \approx G^{-1} = ZZ^{\mathrm{T}}, \tag{22}$$

where Z is upper triangular. Hence ICP with the AINV approximation of K^{-1} and the incomplete decomposition of S can be factorized as follows:

$$\begin{aligned}\widehat{\mathscr{M}}^{-1} &= \begin{bmatrix} I_{3n} & -G^{-1}B^{\mathrm{T}} \\ 0 & I_n \end{bmatrix}\begin{bmatrix} G^{-1} & 0 \\ 0 & -(\tilde{L}\tilde{L}^{\mathrm{T}})^{-1} \end{bmatrix}\begin{bmatrix} I_{3n} & 0 \\ -BG^{-1} & I_n \end{bmatrix} \\ &= \begin{bmatrix} Z & -ZZ^{\mathrm{T}}B^{\mathrm{T}}\tilde{L}^{-T} \\ 0 & \tilde{L}^{-T} \end{bmatrix}\begin{bmatrix} Z^{\mathrm{T}} & 0 \\ \tilde{L}^{-1}BZZ^{\mathrm{T}} & -\tilde{L}^{-1} \end{bmatrix} = \mathscr{U}\mathscr{L},\end{aligned} \tag{23}$$

where $\mathscr{U}$ and $\mathscr{L}$ are upper and lower triangular matrices, respectively. The factorized form (23) of $\widehat{\mathscr{M}}^{-1}$ is very well suited to an efficient implementation.

3.2. Numerical algorithms

The application of both $\mathscr{M}^{-1}$ and $\widehat{\mathscr{M}}^{-1}$ requires the explicit knowledge of the Schur complement matrix S (Eq. (19)). Forming the Schur complement may be time and memory consuming, S being the result of two sparse matrix–matrix products and one sparse sum of matrices. However, it should be noted that the evaluation of $S_0 = BG^{-1}B^{\mathrm{T}}$, which involves the main computational burden in building S, is independent of the time step Δt, and therefore can be done just once at the beginning of the simulation.

Since S is generally less sparse than $\mathscr{A}$, the efficiency of its storage can be increased by dropping the terms below a user-specified tolerance τ_{S}. This can be done only with ICP because dropping some terms of S introduces a new approximation.

The new complete algorithms (Algorithms 1 and 2) for the transient solution of a coupled consolidation problem with *nstep* Δt values using ECP and ICP, respectively, are

Input: τ_A, τ_{CG}

1. Compute an approximate inverse of K: $G^{-1} = ZZ^T$ with drop tolerance τ_A
2. Compute $W = BZ$
3. Compute $S_0 = WW^T$
4. DO $i = 1, nstep$
   (a) Compute $C = \theta\Delta t_i H + P$ and $S = S_0 + C$
   (b) Solve $\mathscr{A}\boldsymbol{x}^{(i)} = \boldsymbol{b}$ by PCG preconditioned with $\mathscr{M}^{-1}$
   with exit test: $\|\boldsymbol{r}\| \leq \tau_{CG}\|\boldsymbol{b}\|$

END DO

Fig. 1. Algorithm 1: solution to the coupled consolidation problem by PCG preconditioned with ECP.

Input: τ_A, τ_S, τ_I, τ_{BCG}

1. Compute an approximate inverse of K: $G^{-1} = ZZ^T$ with drop tolerance τ_A
2. Compute $W = BZ$
3. Compute $S_0 = WW^T$ with drop tolerance τ_S
4. DO $i = 1, nstep$
   (a) Compute $C = \theta \Delta t_i H + P$ and $S = S_0 + C$
   (b) Compute an incomplete factorization of S: $S \approx \tilde{L}\tilde{L}^T$ with tolerance τ_I
   (c) Solve $\mathscr{A}\boldsymbol{x}^{(i)} = \boldsymbol{b}$ by Bi-CGSTAB preconditioned with $\widehat{\mathscr{M}}^{-1}$
   with exit test: $\|\boldsymbol{r}\| \leq \tau_{BCG}\|\boldsymbol{b}\|$

END DO

Fig. 2. Algorithm 2: solution to the coupled consolidation problem by Bi-CGSTAB preconditioned with ICP.

1. $\boldsymbol{v} = Z^T\boldsymbol{r}_1$
2. $\boldsymbol{w} = W\boldsymbol{v} - \boldsymbol{r}_2$
3. solve $S\boldsymbol{y}_2 = \boldsymbol{w}$
4. $\boldsymbol{z} = \boldsymbol{v} - W^T\boldsymbol{y}_2$
5. $\boldsymbol{y}_1 = Z\boldsymbol{z}$

Fig. 3. Algorithm 3: application of ECP in the PCG iteration.

shown in Figs. 1 and 2. Note that with ECP step 4b of Algorithm 2 is skipped and PCG is used instead of Bi-CGSTAB. Moreover, τ_A is the only parameter controlling the ECP performance, while with ICP τ_S and τ_I are also to be provided. Steps 1–3 in both algorithms are independent of Δt and will be referred to as "preprocessing" in the sequel.

In Algorithm 1, the actual solution to $\mathscr{A}\boldsymbol{x} = \boldsymbol{b}$ is accomplished in step 4b. The application of $\mathscr{M}^{-1}$ in the PCG scheme is described by Algorithm 3 (Fig. 3) implementing Eqs. (17) and (18). The most expensive task is performed in step 3 where the solution to a SPD system is needed. This can be efficiently done by using the PCG scheme preconditioned with the incomplete Cholesky decomposition, thus defining an inner PCG cycle within the outer PCG iteration.

1. $\boldsymbol{v} = Z^T\boldsymbol{r}_1$
2. $\boldsymbol{w} = W\boldsymbol{v} - \boldsymbol{r}_2$
3. solve $\tilde{L}\boldsymbol{m} = \boldsymbol{w}$
4. solve $\tilde{L}^T\boldsymbol{y}_2 = \boldsymbol{m}$
5. $\boldsymbol{z} = \boldsymbol{v} - W^T\boldsymbol{y}_2$
6. $\boldsymbol{y}_1 = Z\boldsymbol{z}$

Fig. 4. Algorithm 4: application of ICP in the Bi-CGSTAB iteration.

In Algorithm 2 the actual solution to $\mathscr{A}\boldsymbol{x} = \boldsymbol{b}$ is accomplished in step 4c. Here the standard Bi-CGSTAB method has to be modified to implement the efficient application of $\widehat{\mathscr{M}}^{-1}$. As the preconditioner is known in the factorized form (23), i.e. $\widehat{\mathscr{M}}^{-1} = \mathscr{U}\mathscr{L}$, we can use the so-called "split" preconditioning technique which proves most effective. Based on Eq. (23), Fig. 4 shows the steps needed to compute $\boldsymbol{y} = \widehat{\mathscr{M}}^{-1}\boldsymbol{r}$ (Algorithm 4).

In the sequel with ECP and ICP performance we refer to the performance of PCG preconditioned with $\mathscr{M}^{-1}$ (Algorithms 1 and 3) and Bi-CGSTAB preconditioned with $\widehat{\mathscr{M}}^{-1}$ (Algorithms 2 and 4), respectively. The terms AINV and Jacobi are related to the choice of G^{-1}, while IC(0) and ILLT to the approximation of S.

4. Test problem

A vertical cross-section of the cylindrical porous volume used as a test problem is shown in Fig. 5. The medium consists of a sequence of alternating sandy and clayey layers, with the hydraulic conductivity $k_{\text{sand}} = 10^{-5}$ m/s and $k_{\text{clay}} = 10^{-8}$ m/s, the Poisson ratio $\nu = 0.25$, and the Young modulus $E = 833.33$ MPa, corresponding to a uniaxial vertical compressibility $c_M = 10^{-3}$ MPa^{-1}. Standard Dirichlet conditions are prescribed, with fixed outer and bottom boundaries, and zero pore pressure variation on the top and outer surfaces (see Fig. 5). The second-order Crank–Nicolson finite difference scheme is used ($\theta = 0.5$), with a variable time step Δt_i, $i = 1, \ldots, nstep$.

The sample problem is solved using fully 3-D grids made of linear tetrahedral elements. The pressure and displacement components are discretized with equal-order basis functions. In the first test case, denoted as `M3Dsm`, the grid is generated by projecting a plane triangulation made of

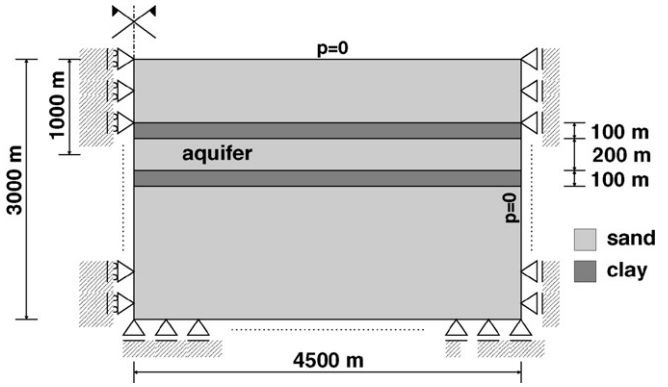

Fig. 5. Schematic representation of a vertical cross-section of the stratified porous medium used as a test problem.

209 nodes and 400 triangles onto 17 layers located at different depths [33]. The grid `M3Dsm` totals $n = 3553$ nodes with a global matrix size N equal to 14,212. This grid is then used to generate a severely ill-conditioned problem (`M3Dsm_l`) by changing the values of k_{clay} to 10^{-11} m/s and c_M to 10^{-2} MPa^{-1} [2].

In the second test case, denoted as `M3D`, a plane triangulation made of 1025 nodes and 2016 triangles is projected onto 31 layers. The `M3D` problem totals $n = 31{,}775$ nodes with $N = 127{,}100$.

4.1. Numerical results

The ECP and ICP performance is compared to that of the ILUT preconditioner with optimal fill-in applied to the native LSL scaled and reordered system [7] for the test cases described above. The LSL-ILUT preconditioning provides an incomplete triangular decomposition of the unsymmetric LSL scaled matrix $\mathscr{A}'$. The fill-in degree is controlled by two user-specified parameters ρ (maximum number of non-zeroes stored for each row) and τ_I (drop tolerance). Their optimal values, i.e. the couple (ρ, τ_I) giving the best solver performance, have to be found empirically via a trial-and-error procedure. The LSL-ILUT has proved a robust and efficient preconditioner for Bi-CGSTAB in the iterative solution to FE coupled consolidation equations [6,7].

All the iterations are completed for a final solution $\boldsymbol{x}$ satisfying the relative error ε

$$\varepsilon = \frac{\|\boldsymbol{x} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} \leqslant 10^{-5}, \tag{24}$$

$\boldsymbol{x}^*$ being a prescribed test solution with all components equal to 1. The experiments are performed on a Compaq DS20 equipped with an alpha-processor "ev6" at 500 MHz, 1.5 GB of core memory, and 8 MB of secondary cache. We use the pure Fortran 90 version of the code compiled with the f90 compiler and `-O4 -tune=ev6 -arch=ev6` options.

Table 1
Problem `M3Dsm`: CPU time (s) for Bi-CGSTAB preconditioned with optimal LSL-ILUT

Δt	# iterations	CPU time [s]		
		Preconditioner	Bi-CGSTAB	Total
10^0	155	8.32	13.75	22.07
10^1	163	7.95	12.04	19.99
10^2	162	7.64	12.13	19.77
10^3	155	6.00	12.83	18.83
10^4	150	3.94	13.13	17.07

The number of Bi-CGSTAB iterations are given.

4.1.1. Test case `M3Dsm`

Table 1 provides the Bi-CGSTAB performance for different Δt values using the LSL-ILUT preconditioner with optimal parameters. The CPU times refer to one solution of system (11) for a given Δt. Note that, as expected [2], both the cost for computing the preconditioner and the total cost decrease as Δt increases because the conditioning of $\mathscr{A}$ improves with Δt. These results are used as a benchmark against PCG preconditioned with ECP and Bi-CGSTAB preconditioned with ICP.

Using ECP with $G^{-1} = \mathrm{diag}(K)^{-1}$ (Jacobi ECP), PCG converges very slowly with ε equal to about 10^{-1} after 10,000 iterations. Although each iteration is very cheap, the slow convergence due to the poor G^{-1} quality precludes its use as a preconditioner.

Table 2 summarizes the performance of ECP with $G^{-1} = ZZ^T$ (AINV ECP) and $\tau_A = 0.05$. The preprocessing time includes the cost for computing G^{-1} and S_0, while the preconditioner CPU time is actually the cost for computing the preconditioner of the inner system (18). The inner PCG iteration is completed with a relative residual equal to 10^{-4}, i.e. 4–5 inner iterations per single outer iteration suffice to solve (18) at the required accuracy. As is known from theory, a high inner accuracy is not really needed for the outer PCG to converge. Comparison with Table 1 reveals that LSL-ILUT is superior to AINV ECP. While AINV ECP yields a reduction of the iteration count, providing evidence of the better conditioning of the iteration matrix $\mathscr{M}^{-1}\mathscr{A}$, the solution to the inner system (18) at each outer iteration turns out to be too expensive.

Table 2
Problem `M3Dsm`: CPU time (s) for PCG preconditioned with AINV ECP

Δt	# iterations outer (inner)	CPU time [s]		
		Preconditioner	PCG	Total
10^0	112 (460)	5.17	31.79	37.37
10^1	112 (464)	5.13	29.05	34.57
10^2	107 (441)	5.09	24.70	30.14
10^3	113 (465)	5.09	31.38	36.80
10^4	114 (467)	4.94	29.12	34.47

The inner solution is obtained with PCG preconditioned with the incomplete Cholesky decomposition. AINV is computed setting $\tau_A = 0.05$.
Preprocessing AINV ECP: 5.28 s.

Table 3
Problem M3Dsm: CPU time (s) for Bi-CGSTAB preconditioned with AINV-ILLT ICP and AINV-IC(0) ICP

	Δt	# iterations	CPU time [s]		
			Preconditioner	Bi-CGSTAB	Total
AINV-ILLT ICP	10^0	68	13.65	7.11	21.08
	10^1	67	11.92	6.18	18.38
	10^2	70	9.16	5.99	15.40
	10^3	71	6.13	5.37	11.73
	10^4	71	3.58	4.49	8.26
AINV-IC(0) ICP	10^0	70	2.83	6.99	10.19
	10^1	66	2.63	6.22	9.16
	10^2	69	2.16	6.35	8.80
	10^3	74	1.52	6.02	7.80
	10^4	72	1.05	4.96	6.02

AINV is computed with $\tau_A = 0.05$, S with $\tau_S = 10^{-4}$, and ILLT with $\tau_I = 10^{-3}$.
Preprocessing AINV-ILLT and AINV-IC(0) ICP: 5.28 s.

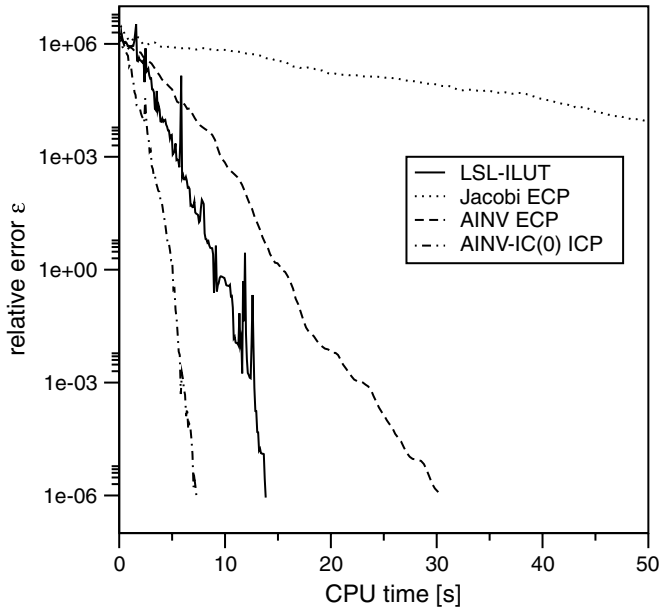

Fig. 6. Problem M3Dsm: Convergence profiles of the relative error vs. the CPU time.

Table 3 shows the ICP performance using AINV ($\tau_A = 0.05$) and both ILLT ($\tau_I = 10^{-3}$) and IC(0) for approximating S^{-1} ($\tau_S = 10^{-4}$). As outlined earlier, PCG is not theoretically guaranteed to converge, hence Bi-CGSTAB is used. In both cases we obtain an improvement in terms of number of iterations and CPU time as compared to Table 1. Note that ICP with AINV-IC(0) yields a speed-up larger than 2 with respect to LSL-ILUT for any time step value.

The convergence profiles ε vs. the solver CPU time for various preconditioners and $\Delta t = 10^0$ are shown in Fig. 6. Notice the smooth PCG convergence (with ECP) as compared to the erratic one of Bi-CGSTAB (with LSL-ILUT and, to a lesser extent, ICP). However, the cost per (outer) iteration of the former turns out to be too high to make it competitive with the latter.

Table 4
Problem M3Dsm_1: CPU time (s) for Bi-CGSTAB preconditioned with AINV-IC(0) ICP

Δt	# iterations	CPU time [s]		
		Preconditioner	Bi-CGSTAB	Total
10^0	96	2.92	10.19	13.87
10^1	94	2.87	8.52	12.12
10^2	96	2.67	9.25	12.63
10^3	92	2.23	8.32	11.39
10^4	99	1.69	9.59	11.86

AINV is computed with $\tau_A = 0.05$ and S with $\tau_S = 10^{-4}$.
Preprocessing AINV-IC(0) ICP: 5.28 s.

4.1.2. Test case M3Dsm_1

This is a very ill-conditioned test case where Bi-CGSTAB preconditioned with LSL-ILUT does not converge for a most standard choice (10^{-2}–10^{-5}) of the drop tolerance τ_I. With smaller τ_I values (10^{-6}–10^{-8}) Bi-CGSTAB converges very slowly satisfying the exit test (24) after about 400 iterations and requiring a CPU time of about 80 s, i.e. much larger than those of the previous tests (Table 1), with the convergence degrading as τ_I decreases. This may occur when the ill-conditioning of $\mathscr{A}$ reflects on an ill-conditioned incomplete decomposition. As a paradoxical result, the conditioning of the incomplete factors can even get worse as the fill-in increases, i.e. for an ILUT preconditioner theoretically approaching $\mathscr{A}^{-1}$. A further explanation of this behavior could be connected with the eigenvalue distribution of the preconditioned matrix. As the ILUT drop tolerance τ_I approaches zero, the preconditioned matrix should approach the identity, i.e. all eigenvalues should approach 1. Therefore, reducing τ_I the real part of the initially negative eigenvalues changes sign. When this happens for a real eigenvalue in relation to

Table 5
Problem M3D: CPU time (s) for Bi-CGSTAB preconditioned with optimal LSL-ILUT and AINV-IC(0) ICP

	Δt	# iterations	CPU time [s]		
			Preconditioner	Bi-CGSTAB	Total
LSL-ILUT	10^0	178	71.81	156.90	231.21
	10^1	185	70.08	169.08	241.66
	10^2	163	63.57	130.44	196.51
	10^3	150	40.05	122.43	165.98
	10^4	79	22.97	67.62	93.08
AINV-IC(0) ICP	10^0	234	12.66	213.61	229.17
	10^1	230	12.81	216.37	231.87
	10^2	227	12.92	206.32	222.03
	10^3	239	12.30	222.03	236.96
	10^4	251	11.38	232.47	246.37

AINV is computed with $\tau_A = 0.1$ and S with $\tau_S = 10^{-4}$.
Preprocessing AINV-IC(0) ICP: 20.12 s.

some specific τ_I the preconditioned system becomes singular.

By contrast, AINV-IC(0) ICP with $\tau_A = 0.05$ and $\tau_S = 10^{-4}$ was successful with the performance shown in Table 4. Notice the fast ICP convergence with the total CPU times only slightly larger than in the M3Dsm test case.

4.1.3. Test case M3D

Table 5 shows the performance of optimal LSL-ILUT compared to AINV-IC(0) ICP, which provides the best result with $\tau_A = 0.1$ and $\tau_S = 10^{-4}$. As can be also seen from the previous results, ICP appears to be less sensitive than LSL-ILUT to the Δt size. In this case for small time steps ICP is comparable or a little better than LSL-ILUT, while it appears to be less efficient for large time steps. This is more connected with the relatively good problem conditioning rather than its size and provides evidence that ICP can be viewed as an effective and robust alternative to LSL-ILUT especially in ill-conditioned problems.

4.1.4. Long term simulation

One major drawback of ICP is perhaps the need for the preprocessing of S_0. Such an additional cost is, however, made up for very quickly in long-term simulations. The test cases M3Dsm and M3D are considered with variable time step $\Delta t_i = f \cdot \Delta t_{i-1}$, $i = 1, \ldots, 150$. The initial step size Δt_0 is set to 10^{-1} s and the magnifying factor f to 1.10, so that

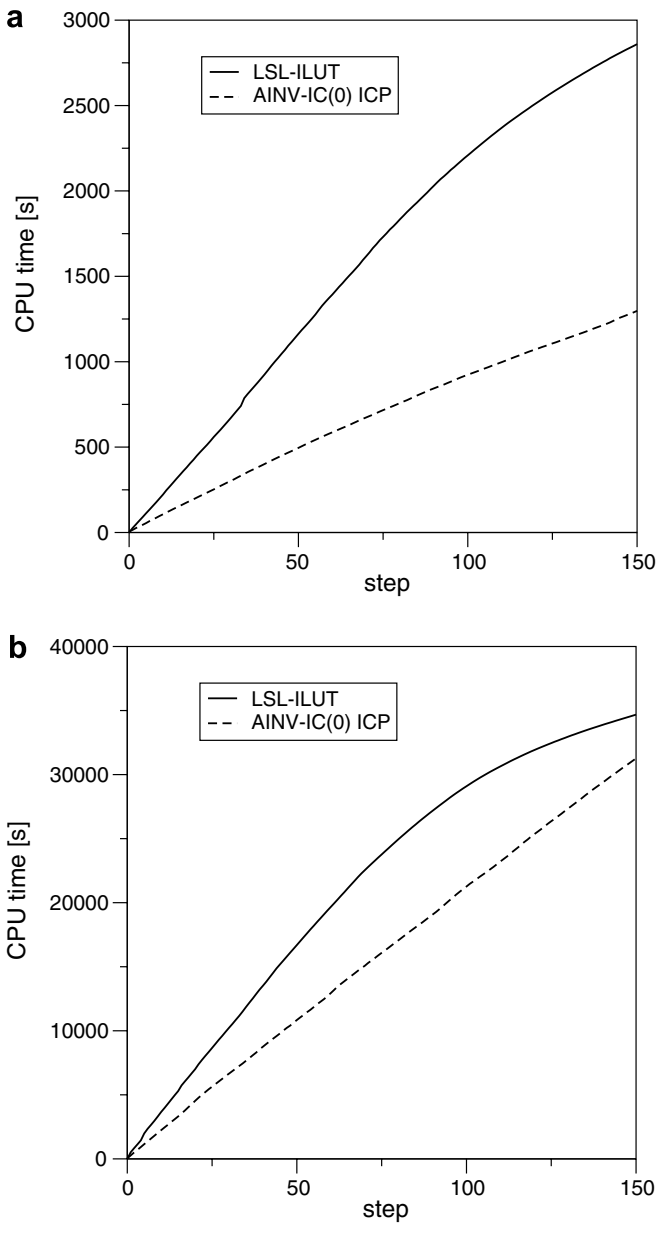

Fig. 7. Long term simulation (150 time steps). Cumulative CPU time (s) to compute the transient solution of problems (a) M3Dsm, (b) M3D.

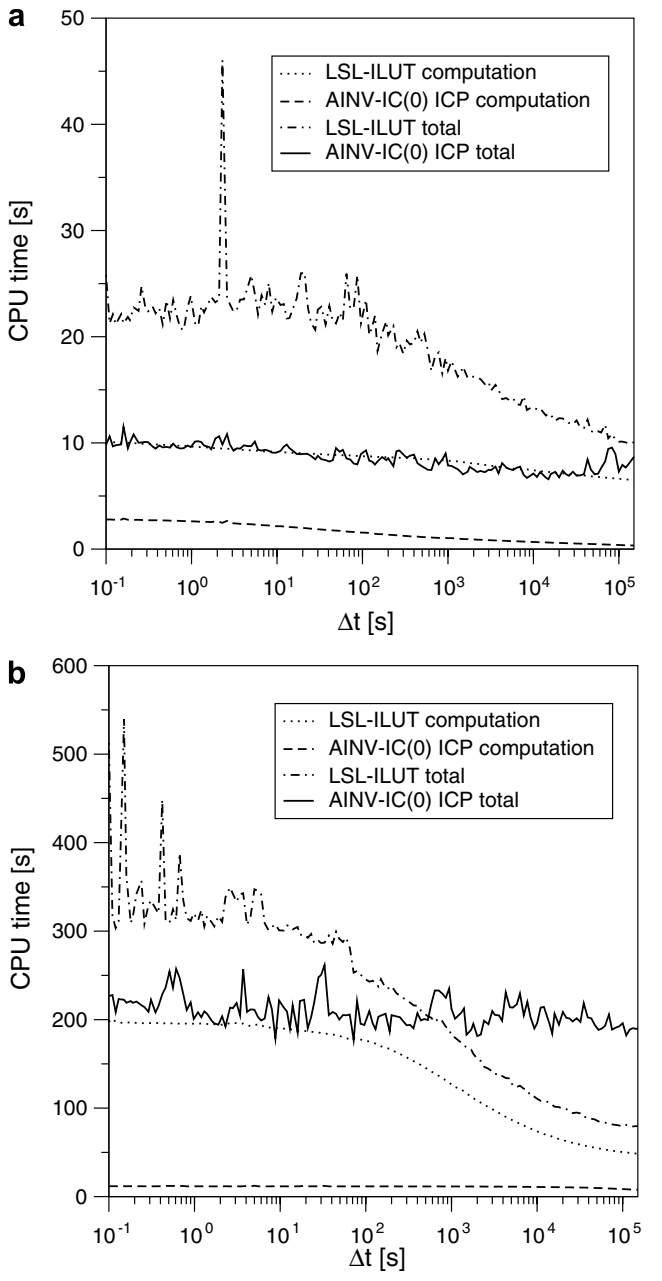

Fig. 8. Long term simulation (150 time steps). CPU time (s) to compute the preconditioner and solve the coupled system vs. the time step size for problems (a) M3Dsm, (b) M3D.

the time step progressively increases up to about 10^5 s as the problem approaches the steady state. Fig. 7 summarizes the LSL-ILUT and AINV-IC(0) ICP total CPU time for running the 150-step transient simulations. Very few steps suffice to make up for the preprocessing cost with a final speed-up of AINV-IC(0) ICP in the `M3Dsm` case larger than 2. The computational cost as the simulation proceeds is shown in detail in Fig. 8, which provides the CPU time vs. Δt required by both the preconditioner computation and the system solution. As is expected, both the costs decrease as Δt increases, but ICP proves less sensitive than LSL-ILUT to the Δt size and more efficient especially in the initial (and more difficult) steps.

It is worth noting that in a long-term simulation the optimal LSL-ILUT parameters are set for the smallest time step and are not changed as Δt increases. This can affect the LSL-ILUT performance for larger Δt, with the preconditioner losing its optimal property. By distinction, the ICP optimal parameters are much less sensitive to Δt with their values generally varying within a pretty small range. For instance, in the cases discussed above typical τ_A values vary between 0.05 and 0.1 and τ_S values between 10^{-3} and 10^{-4} with no significant differences in the algorithm performance over this range.

5. Conclusions

A novel block ICP (Inexact Constraint Preconditioner) has been developed and implemented into projective conjugate gradient-like methods for the efficient iterative solution to FE coupled consolidation equations. ICP is based on the factorized AINV preconditioning of the structural stiffness matrix K and the incomplete Cholesky decomposition of the Schur complement S. A comparison with Bi-CGSTAB preconditioned with optimal LSL-ILUT is made on two realistic 3D consolidation problems. The results can be summarized as follows:

- in the smaller test case ICP is more cost effective with a speed-up larger than 2, while in the larger example the two preconditioners behave similarly;
- ICP is successfully tested for robustness in a severely ill-conditioned problem where LSL-ILUT does not allow for Bi-CGSTAB to converge;
- long-term transient simulations show that the preprocessing cost needed for the AINV and the Schur complement computation is readily compensated within a few initial time steps;
- the ICP parameters τ_A and τ_S prove quite insensitive to the Δt size and typically fall within a limited range of variation;
- as anticipated from theory, ECP (Exact Constraint Preconditioner) allows for PCG to converge with the indefinite FE coupled consolidation system. Though theoretically elegant and attractive, ECP turns out to be computationally less efficient than Bi-CGSTAB implemented with either ICP or LSL-ILUT, because of the larger cost per iteration required by the ECP application.

On summary, the present analysis shows that ICP is an efficient, robust and reliable preconditioner for the iterative solution to FE coupled consolidation models and possesses a promising potential for its application in real problems.

Acknowledgement

This study has been supported by the Italian MIUR project ''Numerical models for multiphase flow and deformation in porous media''.

References

[1] M.A. Biot, General theory of three-dimensional consolidation, J. Appl. Phys. 12 (1941) 155–164.
[2] M. Ferronato, G. Gambolati, P. Teatini, Ill-conditioning of finite element poroelasticity equations, Int. J. Solids Struct. 38 (2001) 5995–6014.
[3] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 631–644.
[4] S.H. Chan, K.K. Phoon, F. Lee, A modified Jacobi preconditioner for solving ill-conditioned Biot's consolidation equations using symmetric quasi-minimal residual method, Int. J. Numer. Anal. Methods Geomech. 25 (2001) 1001–1025.
[5] G. Gambolati, G. Pini, M. Ferronato, Numerical performance of projection methods in finite element consolidation models, Int. J. Numer. Anal. Methods Geomech. 25 (2001) 1429–1447.
[6] G. Gambolati, G. Pini, M. Ferronato, Direct, partitioned and projected solution to finite element consolidation models, Int. J. Numer. Anal. Methods Geomech. 26 (2002) 1371–1383.
[7] G. Gambolati, G. Pini, M. Ferronato, Scaling improves stability of preconditioned CG-like solvers for FE consolidation equations, Int. J. Numer. Anal. Methods Geomech. 27 (2003) 1043–1056.
[8] K.C. Toh, K.K. Phoon, S.H. Chan, Block preconditioners for symmetric indefinite linear systems, Int. J. Numer. Methods Engrg. 60 (2004) 1361–1381.
[9] X. Chen, K.C. Toh, K.K. Phoon, A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations, Int. J. Numer. Methods Engrg. 65 (2006) 785–807.
[10] Y. Saad, ILUT: a dual threshold incomplete ILU factorization, Numer. Linear Algebra Appl. 1 (1994) 387–402.
[11] T. Washio, T. Hishada, H. Watanabe, H. Tezduyar, A robust preconditioner for fluid–structure interaction problems, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4027–4047.
[12] D. Silvester, A.J. Wathen, Fast iterative solution of stabilised Stokes systems. Part II: Using general block preconditioners, SIAM J. Numer. Anal. 31 (1994) 1352–1367.
[13] C. Keller, N.I.M. Gould, A.J. Wathen, Constraint preconditioning for indefinite linear systems, SIAM J. Matrix Anal. Appl. 21 (2000) 1300–1317.
[14] L. Lukšan, J. Vlček, Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems, Numer. Linear Algebra Appl. 5 (1998) 219–247.
[15] I. Perugia, V. Simoncini, Block-diagonal and indefinite symmetric preconditioners for mixed finite elements formulations, Numer. Linear Algebra Appl. 7 (2000) 585–616.
[16] L. Bergamaschi, J. Gondzio, G. Zilli, Preconditioning indefinite systems in interior point methods for optimization, Comput. Optim. Appl. 28 (2004) 149–171.

[17] D.J. Silvester, H.C. Elman, D. Kay, A.J. Wathen, Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow, J. Comput. Appl. Math. 128 (2001) 261–279.
[18] H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations, Numer. Math. 90 (2002) 665–688.
[19] M. Benzi, G. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[20] K. van der Knaap, Nonlinear behavior of elastic porous media, Petroleum Trans. AIME 216 (1959) 179–187.
[21] J. Geertsma, Land subsidence above compacting oil and gas reservoirs, J. Petrol. Technol. 25 (1973) 734–744.
[22] J.R. Booker, J.C. Small, An investigation of the stability of numerical solutions of Biot's equations of consolidation, Int. J. Solids Struct. 11 (1975) 907–917.
[23] L. Bergamaschi, M. Ferronato, G. Gambolati, Efficient preconditioners for Krylov subspace methods in the solution of coupled consolidation problems, in: B.H.V. Topping, G. Montero, R. Montenegro (Eds.), Proceedings of the Fifth International Conference on Engineering Computer Technology, Civil-Comp Press, 2006, pp. 1–14, Paper 84.
[24] A. Greenbaum, Iterative methods for solving linear systems, SIAM Frontiers in Applied Mathematics, Philadelphia (PA), 1997.
[25] M. Rozložník, V. Simoncini, Krylov subspace methods for saddle point problems with indefinite preconditioning, SIAM J. Matrix Anal. Appl. 24 (2002) 368–391.
[26] J. Castro, A specialized interior point algorithm for multicommodity network flows, SIAM J. Optim. 10 (2000) 852–877.
[27] L. Bergamaschi, J. Gondzio, M. Venturin, G. Zilli, Inexact constraint preconditioners for linear systems arising in interior point methods, Comput. Optim. Appl., in press, doi:10.1007/s10589-06-9001-0.
[28] M.J. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses, SIAM J. Sci. Comput. 18 (1997) 838–853.
[29] M. Benzi, M. Tůma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, SIAM J. Sci. Comput. 19 (1998) 968–994.
[30] M. Benzi, K. Cullum, M. Tůma, Robust approximate inverse preconditioning for the conjugate gradient method, SIAM J. Sci. Comput. 22 (2000) 1318–1332.
[31] L.Y. Kolotilina, Y. Yeremin, Factorized sparse approximate inverse preconditionings I. Theory, SIAM J. Matrix Anal. Appl. 14 (1993) 45–58.
[32] M. Benzi, M. Tůma, A comparative study of approximate inverse preconditioners, Appl. Numer. Math. 30 (1999) 305–340.
[33] G. Gambolati, G. Pini, T. Tucciarelli, A 3-D finite element conjugate gradient model of subsurface flow with automatic mesh generation, Adv. Water Resour. 9 (1986) 34–41.