

RMCP: Relaxed Mixed Constraint Preconditioners for Saddle Point Linear Systems arising in Geomechanics

Luca Bergamaschi and Ángeles Martínez

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

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 a Relaxed Mixed Constraint Preconditioner (RMCP) which is a generalization based on a parameter ω of the Mixed Constraint Preconditioner (MCP) developed in [7]. Choice of optimal ω is driven by the spectral distribution of suitable symmetric positive definite (SPD) matrices. Numerical tests performed on realistic 3D problems reveal that RMCP accelerates Krylov subspace solvers by a factor up to three with respect to MCP.

Key words: coupled consolidation, saddle point linear systems, constraint preconditioners, iterative methods

1 Introduction

The time-dependent displacements and fluid pore pressure in porous media are controlled by the consolidation theory. This was first mathematically described by Biot [17], 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.

* Corresponding author: tel. +39 049 8271332, fax. +39 049 8271333
Email addresses: `luca.bergamaschi@unipd.it` (Luca Bergamaschi),
`angeles.martinez@unipd.it` (and Ángeles Martínez).

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 whose solution is addressed by an appropriate time marching scheme. A major computational issue is the repeated solution in time of the resulting discretized indefinite equations, which can be generally written as

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \quad \text{where} \quad \mathcal{A} = \begin{bmatrix} K & B^\top \\ B & -C \end{bmatrix}. \quad (1)$$

Both the sub-matrices K and C are symmetric positive definite (SPD). Denoting with m the number of FE nodes, $C \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times n}$, and $K \in \mathbb{R}^{n \times n}$, where n is equal to $2m$ or $3m$ according to the spatial dimension of the problem if the same interpolation is used for displacement and pressure variables.

Similar problems can be encountered in many fields such as constrained optimization, least squares, coupled consolidation problems and Navier-Stokes equations to mention a few (see [3] for a review of such applications). Iterative solution is recommended against direct factorization methods due to the extremely large size of these systems. However, well established iterative methods such as Krylov subspace methods are very slow or even fail to converge if not conveniently preconditioned.

To accelerate Krylov solvers in the solution of saddle point problems the so-called ‘‘Constraint Preconditioners’’ have been first introduced in constrained optimization [21]. This terminology has been preserved in other fields as well, including least squares and also Navier-Stokes equations [10,24,26,?,28]. For a thorough review of the constraint preconditioning see also [3] and references therein. Constraint preconditioners are written as the inverse of a matrix whose non diagonal blocks are the same as those in \mathcal{A} . It has been proved [21] that the eigenvalues of the preconditioned matrix are all real and positive. However, their application may be very costly since it requires the solution of a linear system at each iteration with an appropriate Schur complement S as the coefficient matrix. A computationally efficient variant of constraint preconditioners is represented by ‘‘Inexact Constraint Preconditioners’’ (ICP) which are based on an approximation to S (or to S^{-1}) by means e.g. of an incomplete Cholesky factorization (or of a sparse approximate inverse). The application of ICP is cheaper with respect to the Constraint Preconditioner. The price to be paid is that the eigenvalues of the preconditioned matrix are no longer all real. An exhaustive analysis of spectral properties of ICP together with development of eigenvalue bounds are performed in [5]. ICP has been proved much more robust and performing than ILUT preconditioners with variable fill-in, computed on the whole saddle point matrix, in [6], where a number of realistic coupled consolidation problems have been solved by both approaches.

In this paper we propose a development of the Mixed Constraint Preconditioners (MCP) introduced in [7] and successfully compared in [8] with other block preconditioners. MCP is nothing but ICP where two approximations for K (P_K and \widetilde{P}_K) and an approximation (P_S) of a suitable Schur complement matrix $S = B\widetilde{P}_K^{-1}B^\top + C$ are available. The MCP is defined as \mathcal{M}^{-1} where

$$\mathcal{M} = \begin{bmatrix} I & 0 \\ BP_K^{-1} & I \end{bmatrix} \begin{bmatrix} P_K & 0 \\ 0 & -P_S \end{bmatrix} \begin{bmatrix} I & P_K^{-1}B^\top \\ 0 & I \end{bmatrix} = \begin{bmatrix} P_K & B^\top \\ B & BP_K^{-1}B^\top - P_S \end{bmatrix}. \quad (2)$$

We propose in this paper a family of relaxed MCP (RMCP) denoted by $\mathcal{M}^{-1}(\omega)$, where ω is a real acceleration parameter and

$$\mathcal{M}(\omega) = \begin{bmatrix} I & 0 \\ BP_K^{-1} & I \end{bmatrix} \begin{bmatrix} P_K & 0 \\ 0 & -\omega P_S \end{bmatrix} \begin{bmatrix} I & P_K^{-1}B^\top \\ 0 & I \end{bmatrix} = \begin{bmatrix} P_K & B^\top \\ B & BP_K^{-1}B^\top - \omega P_S \end{bmatrix}. \quad (3)$$

The aim of this paper is to give a detailed spectral analysis of RMCP, showing that the optimal values of ω is strictly related to extremal (real) eigenvalues of $P_S^{-1}S$ and $P_K^{-1}K$ which can be estimated without affecting the overall efficiency of the iterative method. Numerical results obtained on realistic consolidation problems of very large size show that RMCP may improve MCP performance up to a factor three.

To test the effectiveness of the proposed preconditioner, we also used the RMCP in the acceleration of BiCGSTAB for the solution of saddle point linear system arising from the Mixed Finite Element (MFE) discretization of the Darcy's law in porous media. Even though the spectral properties of the block matrices are significantly different from those of the consolidation problem, we obtained an important reduction of the overall CPU time by properly setting the parameter ω .

The paper is organized as follows. In §2 we characterize the coupled consolidation problem while in §3 we give bounds on extremal eigenvalues of $\mathcal{M}(\omega)^{-1}\mathcal{A}$. In §4 the RMCP sequential and parallel implementations are described; in §5 we provide a description of the test cases. In §6 we report some numerical results that accounts for the effectiveness of the acceleration provided by RMCP. §7 is devoted to the discussion of the results of the parallel implementation of RMCP on a very large-size test case. The results of the RMCP acceleration in solving a linear system arising from MFE discretization of the fluid flow equation are presented in §8. The conclusions are drawn in §9.

2 Finite Element coupled consolidation equations

The system of partial differential equations governing the 3D coupled consolidation process in fully saturated porous media is derived from the classical Biot's formulation [17] and successive modifications as:

$$(\lambda + \mu) \frac{\partial \epsilon}{\partial i} + \mu \vec{\nabla}^2 u_i = \alpha \frac{\partial p}{\partial i} \quad i = x, y, z \quad (4)$$

$$\frac{1}{\gamma} \text{div} \cdot (\mathcal{K} \vec{\nabla} p) = [\phi \beta + c_{br}(\alpha - \phi)] \frac{\partial p}{\partial t} + \alpha \frac{\partial \epsilon}{\partial t} \quad (5)$$

where c_{br} and β are the volumetric compressibility of solid grains and water, respectively, ϕ is the porosity, \mathcal{K} the medium hydraulic conductivity, ϵ the medium volumetric dilatation, 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, div and $\vec{\nabla}$ are the divergence and gradient operator, respectively, x, y, z are the coordinate directions, t is time, and p and u_i are the incremental pore pressure and the components of incremental displacement along the i -direction, respectively.

Use of FE in space yields a system of first order differential equations which can be integrated by the Crank-Nicolson scheme [20]. The resulting linear system has to be repeatedly solved to obtain the transient displacements and pore pressures. The unsymmetric matrix controlling the solution scheme reads:

$$A = \begin{bmatrix} K/2 & -Q/2 \\ \frac{Q^\top}{\Delta t} & H/2 + \frac{P}{\Delta t} \end{bmatrix} \quad (6)$$

where K, H, P and Q are the elastic stiffness, flow stiffness, flow capacity and flow-stress coupling matrices, respectively. Matrix A can be readily symmetrized by multiplying the upper set of equations by 2 and the lower set by $-\Delta t$, thus obtaining the sparse 2×2 block symmetric indefinite matrix (1) where $B = -Q^\top$ and $C = \Delta t H/2 + P$.

A major difficulty in the repeated solution to system (1) is the likely ill-conditioning of \mathcal{A} caused by the large difference in magnitude between the coefficients of blocks K, B and C . The generic (i, j) element of each matrix is related to the hydro-mechanical properties of the porous medium as follows [20]:

$$\begin{aligned} K_{ij} &\propto E \\ B_{ij} &\propto \sqrt{V} \\ C_{ij} &\propto \Delta t \frac{k}{\gamma} + \phi \beta V \end{aligned}$$

where E is the Young modulus of the porous medium and V a characteristic size of the FE grid. Being C_{ij} related to the time integration step Δt , the ill-conditioning of \mathcal{A} is basically dependent on the Δt size. Ferronato et al. [20] have shown that a critical time step Δt_{crit} exists that can be defined as:

$$\Delta t_{crit} = \chi(\psi) \frac{V\gamma}{kE} \quad (7)$$

where $\psi = \phi\beta E$ and χ is a generally unknown dimensionless factor depending on ψ and the element distortion. For $\Delta t \leq \Delta t_{crit}$ the conditioning of \mathcal{A} suddenly degrades with the solution to (1) difficult to get independently of the solver choice. 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 critical ones, with the convergence expected to improve as the simulation proceeds.

3 Spectral Analysis of $\mathcal{M}(\omega)^{-1}\mathcal{A}$

We first recall the eigenvalue bounds of $\mathcal{M}^{-1}\mathcal{A}$.

3.1 Bounds on Eigenvalues of $\mathcal{M}^{-1}\mathcal{A}$

Let P_K and P_S be SPD approximations of K and $S = C + BP_K^{-1}B^\top$, respectively. P_K^{-1} and P_S^{-1} can also be viewed as preconditioners for the corresponding matrices, so that we can define the following SPD preconditioned matrices:

$$K_P = P_K^{-1/2} K P_K^{-1/2} \quad \text{and} \quad S_P = P_S^{-1/2} S P_S^{-1/2}.$$

Let us assume that

$$\begin{aligned} 0 < \alpha_K = \lambda_{\min}(K_P) < 1 < \lambda_{\max}(K_P) = \beta_K, \\ 0 < \alpha_S = \lambda_{\min}(S_P) < 1 < \lambda_{\max}(S_P) = \beta_S, \\ 0 \leq \alpha_C = \lambda_{\min}(\hat{C}) \leq \lambda_{\max}(\hat{C}) = \beta_C, \end{aligned} \quad (8)$$

where $\hat{C} = P_S^{-1/2} C P_S^{-1/2}$. The conditions $1 \in [\alpha_K, \beta_K]$ and $1 \in [\alpha_S, \beta_S]$ are very often fulfilled in practice since preconditioners P_K and P_S are expected to cluster eigenvalues around 1.

In order to characterize the eigenvalues of the preconditioned matrices $\mathcal{M}^{-1}\mathcal{A}$

it is useful to define a matrix \mathcal{P} as

$$\mathcal{P} = \begin{bmatrix} P_K^{-1/2} & 0 \\ 0 & P_S^{-1/2} \end{bmatrix}. \quad (9)$$

The problem of finding the eigenvalues of $\mathcal{M}^{-1}\mathcal{A}$ is therefore equivalent to solving $\mathcal{P}\mathcal{A}\mathcal{P}\mathbf{v} = \lambda\mathcal{P}\mathcal{M}\mathcal{P}\mathbf{v}$. Exploiting the blocks:

$$\mathcal{P}\mathcal{A}\mathcal{P}\mathbf{v} = \lambda\mathcal{P}\mathcal{M}\mathcal{P}\mathbf{v} \longrightarrow \begin{bmatrix} P_K & R^\top \\ R & -\hat{C} \end{bmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \lambda \begin{bmatrix} I & R^\top \\ R & RR^\top - I \end{bmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} \quad (10)$$

where $R = P_S^{-1/2}BP_K^{-1/2}$. The inverse of the right hand side matrix product in (10), can be written as

$$(\mathcal{P}\mathcal{M}\mathcal{P})^{-1} = \begin{bmatrix} I & -R^\top \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ R & -I \end{bmatrix} = \mathcal{U}\mathcal{L}$$

so that the eigenvalues of (10) are the same as those of $\mathcal{L}\mathcal{P}\mathcal{A}\mathcal{P}\mathcal{U}\mathbf{w} = \lambda\mathbf{w}$ which reads:

$$\begin{pmatrix} K_P & (I - K_P)R^\top \\ -R(I - K_P) & R(2I - K_P)R^\top + \hat{C} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}. \quad (11)$$

The (positive) eigenvalues of the projected matrix $K_R = (RR^\top)^{-1}RK_P R^\top$ will also be important in the spectral analysis that follows. It is easy to show that $[\alpha_K^R, \beta_K^R] \subset [\alpha_K, \beta_K]$, where $\alpha_K^R = \lambda_{\min}(K_R)$, and $\beta_K^R = \lambda_{\max}(K_R)$.

The following theorem gives bounds on the eigenvalues of the preconditioned matrix using MCP. We consider separately the cases $C \equiv 0$ and $C \neq 0$. We denote any complex eigenvalue as $\lambda = \lambda_R + i\lambda_I$.

Theorem 1 $C \equiv 0$.

If $\beta_K^R < 2$ then the real eigenvalues of (11) satisfy:

$$\min \left\{ \alpha_K, \frac{\alpha_S}{\beta_K^R} \right\} \leq \lambda \leq \max \{ (2 - \alpha_K^R)\beta_S, \beta_K \}.$$

If $\lambda_I \neq 0$ then

$$\frac{\alpha_K + \alpha_S(2 - \beta_K^R)}{2} \leq \lambda_R \leq \frac{\beta_K + \beta_S(2 - \alpha_K^R)}{2} \quad |\lambda_I| \leq \sqrt{\beta_S} \max\{1 - \alpha_K, \beta_K - 1\}.$$

$$\boxed{C \neq 0}.$$

Let $\beta_K^R < 2$. The real eigenvalues of (11) satisfy:

$$\min \left\{ \alpha_K, \frac{\alpha_S}{\beta_K^R} + \frac{\alpha_C(\beta_K^R - 1)}{\beta_K^R} \right\} \leq \lambda \leq \max\{\beta_K, (2 - \alpha_K^R)\beta_S - \alpha_C(1 - \alpha_K^R)\}. \quad (12)$$

If $\lambda_I \neq 0$ then

$$\frac{\alpha_K + \alpha_S(2 - \beta_K^R) + \alpha_C(\beta_K^R - 1)}{2} \leq \lambda_R \leq \frac{\beta_K + \beta_S(2 - \alpha_K^R) + \alpha_C(1 - \alpha_K^R)}{2}$$

$$|\lambda_I| \leq \sqrt{\beta_S - \alpha_C \max\{1 - \alpha_K, \beta_K - 1\}}. \quad (13)$$

Proof.

See [5]. □

The results contained in Theorem 1 point out that

- (1) Eigenvalues of the preconditioned matrix are clustered around one if those of the preconditioned K and the preconditioned Schur complement are so.
- (2) Matrix C plays an important role to bound eigenvalues of the preconditioned matrix away from zero. The larger the eigenvalues of C (particularly the smallest one), the larger the smallest eigenvalue of the preconditioned matrix.

3.2 Convergence rate

Since the preconditioned matrix is no longer symmetric, it is not possible to give estimates of the convergence rate simply in terms of eigenvalues. A well-known upper bound for the residual norm of a minimum residual iteration such as GMRES involves the condition number of the eigenvector matrix V . Although we do not have theoretical estimates for the condition number of V we experimentally noticed that its value was indeed modest. Thus, we can relate the number of iteration on the ratio between the largest and the smallest eigenvalue of $\mathcal{M}^{-1}\mathcal{A}$ for which Theorem 3.1 gives bounds.

3.3 Bounds on Eigenvalues of RMCP

To develop eigenvalue bounds for RMCP we will use Theorem 3.1, and particularly the results regarding the real eigenvalues of $\mathcal{M}(\omega)^{-1}\mathcal{A}$. The following theorem gives very simple estimates of the eigenvalues of the RMCP preconditioned matrix in terms of ω .

Theorem 2 *Let $\beta_K < 2$ then any real eigenvalue λ of $\mathcal{M}(\omega)^{-1}\mathcal{A}$ satisfies the following bounds:*

$$\min\left\{\alpha_K, \frac{\omega\alpha_S}{2}\right\} \leq \lambda \leq \max\{2\omega\beta_S, \beta_K\}.$$

Moreover the complex eigenvalues satisfy

$$\begin{aligned} \frac{\alpha_K}{2} &\leq \lambda_R \leq \beta_K + \frac{\omega\beta_S}{2} \\ |\lambda_I| &\leq \sqrt{\omega\beta_S} \max\{1, \beta_K - 1\}. \end{aligned}$$

Proof.

From (12) and observing that using RMCP all eigenvalues of S_P are multiplied by ω , we have:

$$\min\left\{\alpha_K, \omega\frac{\alpha_S}{\beta_K^R} + \frac{\alpha_C(\beta_K^R - 1)}{\beta_K^R}\right\} \leq \lambda \leq \max\{\beta_K, \omega(2 - \alpha_K^R)\beta_S - \alpha_C(1 - \alpha_K^R)\}.$$

These bounds can be simplified by using $\alpha_C \geq 0$, $\beta_K^R \leq \beta_K$ and $\alpha_K^R \geq \alpha_K > 0$ thus obtaining

$$\min\left\{\alpha_K, \frac{\omega\alpha_S}{2}\right\} \leq \lambda < \max\{\beta_K, 2\omega\beta_S\}.$$

Regarding complex eigenvalues, using (13) we obtain the bounds:

$$\begin{aligned} \frac{\alpha_K + \omega\alpha_S(2 - \beta_K^R) + \alpha_C(\beta_K^R - 1)}{2} &\leq \lambda_R \leq \frac{\beta_K + \omega\beta_S(2 - \alpha_K^R) + \alpha_C(1 - \alpha_K^R)}{2} \\ |\lambda_I| &\leq \sqrt{\omega\beta_S - \alpha_C} \max\{1 - \alpha_K, \beta_K - 1\}. \end{aligned}$$

Using again $\alpha_C \geq 0$, $\beta_K^R \leq \beta_K$ and $\alpha_K^R \geq \alpha_K > 0$ these bounds simplify to:

$$\begin{aligned} \frac{\alpha_K}{2} &\leq \lambda_R \leq \beta_K + \frac{\omega\beta_S}{2} \\ |\lambda_I| &\leq \sqrt{\omega\beta_S} \max\{1, \beta_K - 1\}. \end{aligned} \tag{14}$$

□

Since convergence speed of our iterative methods preconditioned by RMCP depends in part on ratio between the largest and the smallest real eigenvalues of $\mathcal{M}(\omega)^{-1}\mathcal{A}$ our aim is to find ω which minimizes this ratio:

$$\kappa_R = \frac{R_{\max}}{R_{\min}} \leq \frac{\max\{\beta_K, 2\omega\beta_S\}}{\min\left\{\alpha_K, \frac{\omega\alpha_S}{2}\right\}}, \quad (15)$$

where R_{\max} and R_{\min} denote the largest and the smallest **real** eigenvalue of $\mathcal{M}(\omega)^{-1}\mathcal{A}$. The optimal ω -value depends on accurate knowledge of $\alpha_K, \beta_K, \alpha_S, \beta_S$. It is well known that computing the smallest eigenvalue is a more time consuming procedure, whatever the algorithm employed, as compared to approximating the largest one. The next theorem states that it is sufficient to know β_K, β_S to provide a good approximate value of ω_{opt} . Let us define $c_K = \kappa(K_P)$ and $c_S = \kappa(S_P)$.

Theorem 3 *Let $\omega = \frac{\beta_K}{\beta_S}$, then the ratio between extremal real eigenvalues of the preconditioned matrix is bounded by*

$$\frac{R_{\max}}{R_{\min}} \leq \max\{2c_K, 4c_S\}.$$

Moreover the complex eigenvalues satisfy

$$\begin{aligned} \frac{\alpha_K}{2} \leq \lambda_R \leq \frac{3\beta_K}{2} \\ |\lambda_I| \leq \sqrt{\beta_K} \max\{1, \beta_K - 1\}. \end{aligned}$$

Proof. e

$$\begin{aligned} R_{\max} &\leq \max\{\beta_K, 2\omega\beta_S\} = 2\beta_K; \\ R_{\min} &\geq \min\left\{\alpha_K, \frac{\alpha_S}{2} \frac{\beta_K}{\beta_S}\right\} = \min\left\{\alpha_K, \frac{\beta_K}{2c_S}\right\}. \end{aligned}$$

Hence

$$\frac{R_{\max}}{R_{\min}} \leq \max\left\{\frac{2\beta_K}{\alpha_K}, 2\beta_K \frac{2c_S}{\beta_K}\right\} = \max\{2c_K, 4c_S\}.$$

The bounds for complex eigenvalues are directly derived from (14). □

Remark 1 *In real applications we always found that $c_S < \frac{c_K}{4}$ and $\alpha_S \gg \alpha_K$, that is the preconditioned Schur complement S_P is better conditioned than K_P and the eigenvalues of S_P are shifted with respect to those of K_P . In such a situation, the bound of Theorem 3.3 would give $\kappa_R \leq 2c_K$ which is only twice the optimal value of κ_R obtainable at the price of costly iterations to*

approximate α_K and α_S . A further outcome of the above theorem is that the complex eigenvalues are estimated in terms of extremal eigenvalue of K_P only, the imaginary part being bounded by $|\lambda_I| < \sqrt{\beta_K} \max\{1, \beta_K - 1\}$.

4 Mixed Constraint Preconditioner

4.1 Sequential implementation

The Mixed Constraint Preconditioner (MCP) proposed in [7] is based on two different approximations of the (1,1) block K . The first one, $P_K = L_K L_K^\top$, is obtained by means of an incomplete Cholesky (IC) factorization with fill-in and drop tolerance. The second one provides an approximation of its inverse ($\hat{P}_K^{-1} = Z_K Z_K^\top$), following the AINV approach [4,2], which is needed to explicitly construct the Schur complement matrix. S is then preconditioned by a simple IC(0) preconditioner. In detail

$$\hat{S} = B Z_K Z_K^\top B^\top + C, \quad P_S = L_S L_S^\top.$$

Note that the preconditioned Schur complement

$$S_P = P_S^{-1/2} \left(B (L_K L_K^\top)^{-1} B^\top + C \right) P_S^{-1/2}$$

is the result of two approximation since L_S is the Cholesky factor of an already approximated Schur complement matrix \hat{S} .

The MCP application requires first the explicit calculation of the $\hat{S} = B Z_K Z_K^\top B^\top + C$ and then its incomplete triangular factor. Forming \hat{S} may be time and memory consuming being the result of two sparse matrix-matrix products and one sparse sum of matrices. However, it may be noted that the evaluation of $S_0 = B Z_K Z_K^\top B^\top$, which involves the main computational burden of \hat{S} , is independent of the time step Δt , and therefore can be done just once at the beginning of the simulation. The Relaxed Mixed Constraint Preconditioner takes on the form:

$$\mathcal{M}^{-1} = \begin{bmatrix} L_K^{-\top} & -L_K^{-\top} L_K^{-1} B^\top \hat{L}_S^{-\top} \\ 0 & \hat{L}_S^{-\top} \end{bmatrix} \begin{bmatrix} L_K^{-1} & 0 \\ \hat{L}_S^{-1} B L_K^{-\top} L_K^{-1} & -\hat{L}_S^{-1} \end{bmatrix}, \quad (16)$$

$$\hat{L}_S = \frac{1}{\sqrt{\omega}} L_S.$$

4.2 Parallel FSAI-based MCP

The previously described MCP implementation relies on IC or AINV preconditioners. The first one is, as well known, difficult to parallelize in both the construction and application phase. The AINV preconditioner is suitable to parallelization only in its application phase. We therefore choose to use another sparse approximate inverse preconditioner (FSAI), which has been initially proposed in [22] and [23], and it has been later developed and successfully implemented in parallel by Bergamaschi et al. in [11]. Given any SPD matrix K the FSAI preconditioner approximately factorizes its inverse as a product of two sparse triangular matrices as $K^{-1} \approx W^\top W$. The choice of nonzeros in W are based on a sparsity pattern which in our work is the same as \tilde{K}^d where \tilde{K} is the result of *prefiltration* [14] of K i.e. dropping of all elements below of a threshold parameter δ . The computed W is then sparsified by dropping all the elements which are below a second tolerance parameter (ε). The final FSAI preconditioner is therefore related to the following three parameters: δ , prefiltration threshold; $d = 1, 2, 4$, power of K generating the sparsity pattern; ε , postfiltration threshold.

Recalling equation (16), the FSAI-RMCP can be written directly as:

$$\mathcal{M}(\omega)^{-1} = \begin{bmatrix} W_1^T & -W_1^T W_1 B^\top \widehat{W}_S^T \\ 0 & \widehat{W}_S^T \end{bmatrix} \begin{bmatrix} W_1 & 0 \\ \widehat{W}_S B W_1^T W_1 & -\widehat{W}_S \end{bmatrix}, \quad (17)$$

$$\widehat{W}_S = \frac{1}{\sqrt{\omega}} W_S;$$

where W_1 is a FSAI factor of K ($P_K^{-1} = (W_1 W_1^\top)$) and W_S is the FSAI factor of the approximate Schur complement matrix \tilde{S} , $P_S^{-1} = W_S^\top W_S$. The Schur complement matrix \tilde{S} is evaluated as $\tilde{S} = B W_2^\top W_2 B^\top + C = S_0 + C$, W_2 being the triangular factor of a sparser FSAI approximation of K^{-1} , obtained from W_1 by a further postfiltration. More details on FSAI-MCP algorithms and on its parallel performance can be found in [15,13,12].

The construction of the preconditioner is therefore based on the following parameters:

- (1) δ_1 , d_K and ε_1 , for the 1st FSAI preconditioner (W_1).
- (2) ε_2 , postfiltration threshold for W_2
- (3) δ_S , d_S and ε_S , for the FSAI preconditioner applied to the Schur complement matrix (W_S).

Our parallel version of the RMCP code is written in FORTRAN 90 and exploits the MPI library for exchanging data among the processors. We used a

block row distribution of all matrices, that is, with complete rows assigned to different processors. All matrices involved are stored in static data structures in CSR format.

The FSAI-RMCP preconditioners will be used to accelerate the BiCGSTAB Krylov subspace methods which is essentially based on matrix-vector products. We made use of an optimized parallel matrix-vector product which has been developed in [25] showing its effectiveness up to 1024 processors.

All tests are performed on the IBM SP6/5376 cluster at the CINECA Centre for HPC, equipped with IBM Power6 processors at 4.7 GHz with 168 nodes, 5376 computing cores, and 21 Tbytes of internal network RAM. The code is written in Fortran 90 and compiled with `-O4 -q64 -qarch=pwr6 -qtune=pwr6 -qnoipa -qstrict -bmaxdata:0x70000000` options.

5 Test problems

A vertical cross-section of the cylindrical porous volume used as a test problem is shown in Figure 1. 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 porosity $\phi = 0.20$, 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 Figure 1). The upper boundary is a traction-free plane.

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 209 nodes and 400 triangles onto 17 layers located at different depths. The grid **M3Dsm** totals $n = 3553$ nodes with a global matrix size N equal to 14212.

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 $n = 31775$ nodes with $N = 127100$.

The third test case, **PoRiver**, considers the simulation of the consolidation of a real gas reservoir of the Po Valley, Italy, used for underground gas storage purposes. The reservoir is a complex multi-layer structure consisting of 5 mineralized pools about 1,200-m deep connected to regional active waterdrives with several interbedded clay lenses. The discretized medium has an areal extent of 50×50 km and goes down to 10,000 m depth. The petro-physical

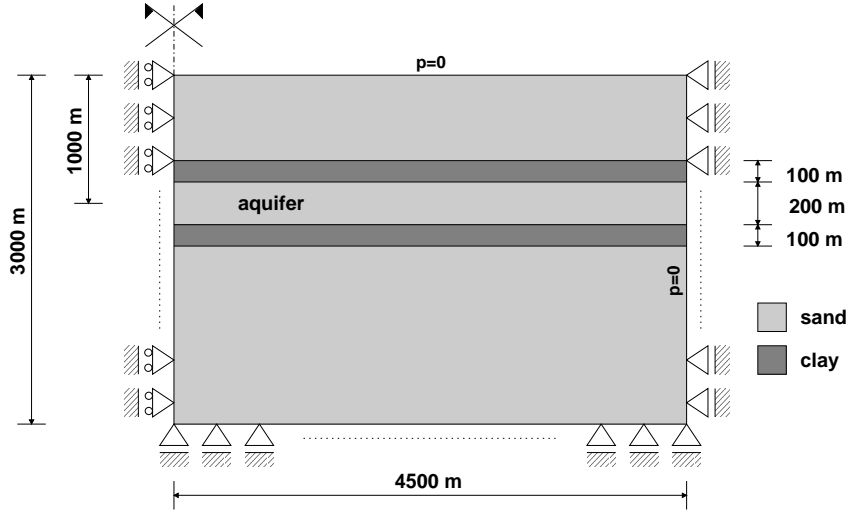


Fig. 1. Schematic representation of a vertical cross-section of the stratified porous medium used as a test problem.

and geomechanical properties of the different layers have been provided by Eni-E&P on the basis of the calibration of the production multi-phase model. The problem is discretized with a 3D tetrahedral grid totaling 299 734 nodes and 1 746 044 elements for 1 198 936 unknowns.

Finally, the LARGE3D test case simulates the compaction of a shallow confined aquifer due to groundwater withdrawal in a representative 3D sedimentary basin at a regional scale. The discretized medium has an areal extent of 20×20 km and consists of an alternating sequence of sand and clay layers down to 5500 m depth, with the hydraulic conductivity $k_{\text{sand}} = 10^{-4}$ m/s and $k_{\text{clay}} = 10^{-7}$ m/s, porosity 0.20 and Poisson ratio 0.30. The mechanical properties of the porous medium vary with depth according to the hypo-plastic law developed in [1] and are representative of the Northern Adriatic sediments, Italy. Dirichlet conditions are prescribed on the bottom boundary and zero pore pressure variation on the rest of the boundary. The problem is discretized with a 3D tetrahedral grid obtained by projecting a 2D mesh made of 11 765 nodes and 23 128 triangles over 46 layers with variable thickness, totaling 541 190 nodes and 3 122 280 elements.

All the test cases are solved using $\Delta t = 1$ which yields the most ill-conditioned linear system.

In Table 1 we summarize the size and nonzeros of the tests described above.

Table 1

Size and nonzeros for the sample matrices.

	N	$\text{nnz}(\mathcal{A})$
M3dsm	12353	707504
M3d	127100	7 426480
PoRiver	1 198936	70 812224
Large3D	2 117700	124 408336

6 Numerical results. Sequential Computations

In this section we present the results of our RMCP preconditioner in combination with the BiCGSTAB Krylov subspace solver in the solution of the first three test cases. In all the runs the BiCGSTAB iteration has been stopped whenever the following exit test on the residual \mathbf{r}_k was satisfied:

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|} < \text{tol},$$

where $\text{tol} = 10^{-12}$ (M3dsm and M3D problems) or $\text{tol} = 10^{-8}$ (PoRiver and Large3D problems). This possibly very low tolerance is required to have in all test cases a relative error of order 10^{-6} . The initial solution has been set to $\mathbf{x}_0 = \mathcal{M}^{-1}\mathbf{b}$.

For each test case we select some values of the parameters which drive the fill-in of the preconditioners for matrices K and S . In detail, the IC preconditioner for K depends on τ_K – the dropping threshold – and lfil – the maximum allowed fill-in per row – while the AINV preconditioner is based on a single threshold parameter τ_Z .

The CPU times (in seconds) refer to running a Fortran 90 code on an IBM Power6 with 4.7 GHz RAM. We denote the relevant CPU times for the comparison as T_p the CPU time needed for computing the preconditioner for the approximated Schur complement, T_{sol} the time for the iterative solver and $T_{tot} = T_p + T_{sol}$. The other time-consuming tasks are: computation of $\text{IC}(\tau_K, \text{lfil})$ preconditioner for K , computation of AINV preconditioner for K and the computation of Schur complement matrix. We note that these last tasks can be carried out at the beginning of the simulation since they do not depend on Δt choice. With $T(L_K)$ we refer to the time to compute the $\text{IC}(\tau_K, \text{lfil})$ preconditioner for block K .

We also provide a measure ρ of the density of the preconditioner matrices as:

$$\rho = \rho_K + \rho_S = \frac{2\text{nnz}(L_K) - n}{\text{nnz}(\mathcal{A})} + \frac{2\text{nnz}(L_S) - m}{\text{nnz}(\mathcal{A})}.$$

Parameter ρ gives an indication of the additional core memory needed for computing and storing the preconditioner.

6.1 Eigenvalue approximation

The key of RMCP success is based on efficient and “cheap” approximation of extreme eigenvalues of K_P and S_P . This is accomplished by the DACG (Deflation-Accelerated Conjugate Gradient) procedure, developed in [9] which has been shown superior to well-known procedures such as ARPACK or Jacobi-Davidson methods in evaluating a few of the smallest eigenvalues of SPD pencils [16]. The DACG procedure has been developed for computing the smallest eigenpairs of $A\mathbf{x} = \lambda B\mathbf{x}$. It is used here also to compute the largest ones by using $B = K_P$ or $B = S_P$ and $A = I$.

6.2 M3d- matrices

We start with the smaller problem M3Dsm in order to show how the eigenvalue distribution changes with ω . We choose the following MCP parameters (see [5]): $\tau_K = \tau_Z = 0.1$ and `lfil`= 10. For this case we found that $\alpha_K = 0.027$, $\beta_K = 2.022$ while $\alpha_S = 0.108$, $\beta_S = 7.390$. Since $\beta_S > \beta_K$ and $\alpha_S > \alpha_K$, ω must be less than one. We therefore run the MCP-BiCGSTAB code for different values of $\omega \in [0.1, 1]$. The results in terms of iteration number and real condition number (κ_R) are displayed in Figure 2.

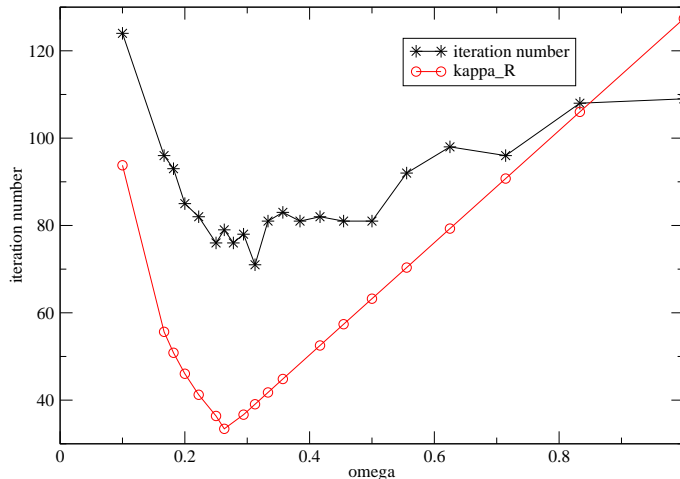


Fig. 2. Number of iterations (stars) and real condition number κ_R (circles) vs ω .

From the figure we can notice that

- (1) There is an optimal value of ω which improves the iteration number obtained with $\omega = 1$. In particular we have 109 iteration with the naive MCP and 71 iteration using $\omega = 0.31$. Note that using Theorem 3 we would obtain $\omega = 0.27$.
- (2) The qualitative plots of $\kappa_R(\omega)$ and $\text{iter}(\omega)$ are much similar. In other words, minimizing the ratio between largest and smallest eigenvalue of $\mathcal{M}^{-1}\mathcal{A}$ yield an ω -value very close to the optimal one.

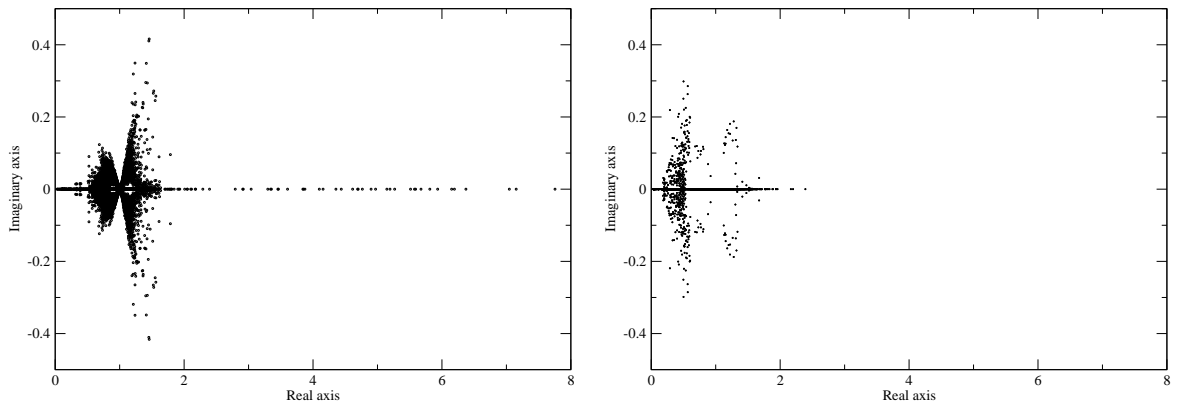


Fig. 3. Eigenvalue distribution of preconditioned matrix with $\omega = 1$ (left) and $\omega = 0.31$ (right).

Figure 3 depicts the two eigenvalue distributions in the complex plane for $\omega = 1$ and $\omega = 0.31$. From the figure we may appreciate the clustering of eigenvalues, both real and complex, around one, using RMCP with $\omega = 0.31$.

The results regarding the M3D matrix are reported in Table 2.

For each combination of the parameters for the (1,1) block and the Schur complement matrix we compare the MCP with RMCP using ω given by Theorem 3.3. The improvement provided by RMCP is not very high, however it holds for every $(\text{lfil}, \tau_K, \tau_Z)$ combination. For this matrix we finally notice that the best MCP combination of parameters produces 59 iterations and 18.5 seconds CPU time whereas the best RMCP provides 50 iterations and 14.1 seconds.

Table 2

Summary results for the M3D matrix.

ω	lfl	τ_K	τ_Z	ρ_K	ρ_S	iter	T_p	T_{sol}	T_{tot}
1	20	0.1	0.1	0.28	1.14	115	4.9	20.3	25.2
1	20	0.1	0.3	0.28	0.45	164	0.7	26.5	27.2
1	30	10^{-2}	0.3	0.64	0.27	109	0.7	19.4	20.1
1	50	10^{-4}	0.05	1.28	1.42	61	18.3	18.4	36.7
1	50	10^{-4}	0.1	1.28	0.45	59	4.5	14.0	18.5
1	50	10^{-4}	0.3	1.28	0.27	88	0.7	20.7	21.4
0.5	20	0.1	0.1	0.28	1.14	106	4.9	18.9	23.9
0.2	20	0.1	0.3	0.28	0.45	114	0.7	18.0	18.7
0.25	30	10^{-2}	0.3	0.64	0.27	65	0.7	11.5	12.2
0.25	50	10^{-4}	0.05	1.28	1.42	30	18.3	8.9	27.2
0.25	50	10^{-4}	0.1	1.28	0.45	50	4.5	9.6	14.1
0.1	50	10^{-4}	0.3	1.28	0.27	71	0.7	16.8	17.5

6.3 Po878 matrix

The results regarding the Po878 matrix are summarized in Tables 3 to 5. We selected three combinations of τ_K , lfl τ_Z parameters. In Table 3 we report the density of preconditioners and extremal real eigenvalues of S_P , K_P and $\mathcal{M}^{-1}\mathcal{A}$ using $\omega = 1$, that is the original MCP.

Table 3

Po878 matrix. MCP with $\omega = 1$. Parameters of the 3 test runs together with extremal eigenvalues of K_P , S_P and ratio κ_R between largest and smallest real eigenvalue of $\mathcal{M}^{-1}\mathcal{A}$.

# run	lfl	τ_K	τ_Z	ρ_K	ρ_S	β_K	α_K	β_S	α_S	κ_R
1	50	10^{-4}	1	1.21	0.14	1.876	0.0112	37.253	0.194	3105
2	30	10^{-4}	1	0.75	0.14	1.875	0.0067	37.255	0.194	5190
3	20	10^{-4}	1	0.51	0.14	1.885	0.0042	37.243	0.195	8195

The results of the MCP runs for the three cases are reported in Table 4. These runs differ in the choice of the parameters for the preconditioner for K . We experimentally found that the optimal threshold value for the AINV preconditioner is $\tau_Z = 1$ i.e. a diagonal approximation of K for the construction of the Schur complement matrix. A more dense AINV preconditioner would yield a highly dense S and an increasing cost for the Schur complement pre-

conditioning without significantly reducing the iteration number. From Table 4 we see that cases #1 and #2 provide the smallest CPU time, indicating that the an efficient IC preconditioner for matrix K must be devised to obtain fast accuracy.

Table 4

CPU times and iteration numbers for MCP ($\omega = 1$) in solving the three test cases of Table 3.

# run	$T(L_K)$	iter	T_p	T_{sol}	T_{tot}
1	151.4	263	3.5	527.8	531.3
2	77.4	315	3.5	520.2	523.7
3	44.9	540	3.5	737.3	740.8

Table 5

CPU times and iteration numbers for RMCP with $\omega = 0.05$ in solving the three test cases of Table 3.

# run	iter	T_p	T_{eig}	T_{sol}	T_{tot}	k_R
1	71	3.5	12.8	145.9	162.2	212
2	102	3.5	11.2	165.5	180.2	259
3	142	3.5	9.4	192.4	205.7	409

To test the effectiveness of our RMCP acceleration we used $\omega = 0.05 \approx \frac{\beta_K}{\beta_S}$ as suggested by Theorem 3. We report in Table 5 the results of these RMCP runs. We also show in this table the preprocessing CPU time to approximate β_K and β_S (T_{eig}) and the value of κ_R .

The RMCP(0.05) preconditioner provides a reduction of more than three times the iteration number and the total CPU time with respect to MCP. This also accounted by the reduction of the ratio between largest and smallest real eigenvalue of $\mathcal{M}^{-1}\mathcal{A}$ (κ_R). Note that the preprocessing time to compute extremal eigenvalues is negligible if compared to the improvement in the number of iterations and CPU time.

We conclude this Section by showing the convergence profile (in Figure 4) of RMCP-BiCGSTAB with $\omega = 1$ and $\omega = 0.05$ for run # 2.

7 Parallel results and scalability

In this section we report the results of our parallel implementation of the RMCP code on the largest `Large3d` problem. We will use a strong scaling measure to see how the CPU times vary with the number of processors for a fixed total problem size. We will denote with T_p the total CPU elapsed

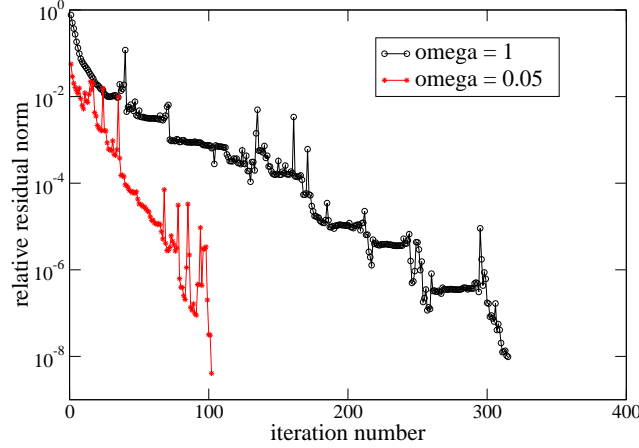


Fig. 4. Convergence profile of RMCP-BiCGSTAB for run # 2.

times expressed in seconds on p processors. As relative measures of the parallel efficiency achieved by the code we denote as $S_p^{(\bar{p})}$ the pseudo speedup computed with respect to the smallest number of processors (\bar{p}) used to solve a given problem and $E_p^{(\bar{p})}$ the corresponding efficiency:

$$S_p^{(\bar{p})} = \frac{T_{\bar{p}}\bar{p}}{T_p}, \quad E_p^{(\bar{p})} = \frac{S_p^{(\bar{p})}}{p} = \frac{T_{\bar{p}}\bar{p}}{T_p p}.$$

Table 6 summarizes the choice of the parameters which have been described in Section 4.1. Their values have been selected in order to give the best performance in terms of CPU time for both MCP and RMCP. As we did in Section 6 we also compute a measure of the density of the preconditioners $\rho = \rho_K + \rho_S$ where now $\rho_K = \frac{2\text{nnz}(W_1) - n}{\text{nnz}(\mathcal{A})}$ and $\rho_S = \frac{2\text{nnz}(W_S) - m}{\text{nnz}(\mathcal{A})}$.

Table 6

Combinations of parameters for the Large3d problem

δ_1	d_K	ϵ_1	ρ_K	ϵ_2	δ_S	d_S	ϵ_S	ρ_S	ρ
0.01	2	0.00	1.14	0.05	0.01	2	0.1	0.24	1.38

The resulting preconditioner is only a little bit more dense than the full saddle point matrix, as accounted for by the value of $\rho = 1.38$.

We present the following timings, all given in seconds: T_{P1} is the preprocessing time needed to construct W_1 , W_2 and S_0 , T_{P2} refers to the construction of W_S and T_{sol} to the CPU time required by the iterative solver. Finally, $T_{tot} = T_{P2} + T_{sol}$ is the total CPU time.

The results regarding the Large3D matrix are summarized in Table 7. Inspection of this table shows that using $\omega = 0.74$, obtained once again from

Table 7

Timings, iteration numbers and pseudo-efficiencies for the Large3D test case for $p=4$ to 512.

p	T_{P1}	T_{P2}	$\omega = 1$				$\omega = 0.74$			
			iter	T_{sol}	T_{tot}	$E_p^{(4)}$	iter	T_{sol}	T_{tot}	$E_p^{(4)}$
4	90.0	10.3	376	610.9	621.6		321	479.3	489.6	
8	46.0	6.7	379	298.7	305.4	1.02	277	226.1	232.8	1.05
16	23.5	4.2	327	136.1	140.3	1.11	274	117.5	121.8	1.00
32	12.5	3.3	359	77.6	80.9	0.96	280	60.9	64.2	0.95
64	6.6	2.5	357	42.0	44.5	0.87	267	29.4	31.9	0.96
128	3.5	1.7	387	22.0	23.7	0.82	268	15.2	16.9	0.91
256	1.9	1.4	433	10.8	12.2	0.80	329	8.2	9.6	0.80
512	1.1	1.1	413	6.2	7.3	0.67	283	4.3	5.4	0.71

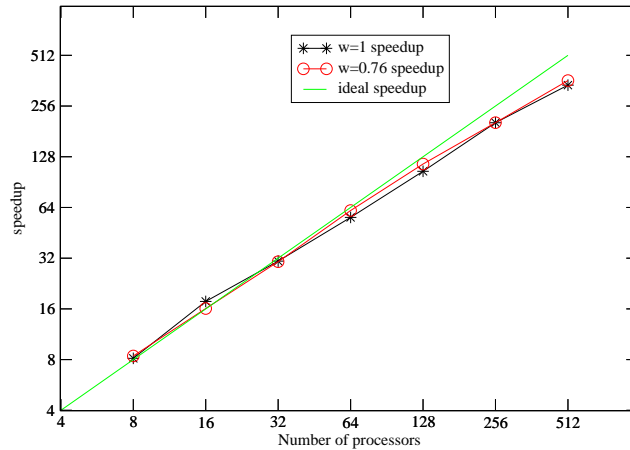


Fig. 5. Pseudo-speedups regarding total CPU times of FSAI-RMCP code using $\omega = 1$ and $\omega = 0.74$.

Theorem 3.3, provides a generalized reduction of the number of iterations and CPU time for the iterative solution, irrespective of the number of processors employed. Note that the preprocessing CPU time to approximate the extremal eigenvalues of P_K and P_S has not been reported, being in all cases less than 5 percent of the total time.

We also notice that the scalability of the two codes are very satisfactory as accounted by the pseudo efficiencies computed in Table 7 and, more evidently,

by Figure 5 where the pseudo-speedups of FSAI-MCP and FSAI-RMCP are shown to be very close to the optimal speedup.

8 Saddle point systems arising from Mixed Finite Elements discretizations

The proposed class of preconditioners can be applied also to other models that give raise to saddle point linear systems. In this Section we use the RMCP in the acceleration of iterative methods for the solution of the saddle point linear systems arising from the Mixed Finite Element (MFE) discretization of fluid flow in porous media.

The fluid mass balance is prescribed by the continuity equation:

$$\operatorname{div} \cdot \vec{v} + \frac{\partial}{\partial t} (\phi \beta p + \alpha) = -\frac{\partial}{\partial t} (\operatorname{div} \cdot \vec{u}) + f \quad (18)$$

where \vec{u} the (known) medium displacements and p the pore pressure; ϕ is the medium porosity, β the fluid compressibility, t time, f a flow source or sink and \vec{v} the Darcy flux. Equation (18) must be coupled with the Darcy law defining \vec{v} :

$$\rho g \mathcal{K}^{-1} \vec{v} + \vec{\nabla} p = 0 \quad (19)$$

with \mathcal{K} the hydraulic conductivity tensor and ρg the fluid specific weight.

Equations (18) and (19) form a coupled partial differential system defined on a 3-D domain Ω bounded by the frontier Γ with \vec{v} and p as unknowns. This system can be solved when appropriate boundary and initial conditions apply.

The fluid pore pressure and Darcy flux are discretized in space with a piecewise constant polynomial and with the lowest order Raviart-Thomas spaces (RT0), respectively, satisfying the LBB condition [27] thus ensuring the well-posedness of the discrete problem. After discretization, a linear system of the form (1) has to be solved at each timestep. Here K is the matrix of the scalar product between RT0 basis functions, B (B^\top) discretizes the gradient (divergence) operator and C is a diagonal matrix depending on Δt . If the steady state problem has to be solved, (or if $\beta = 0$ i.e. the fluid is incompressible) then $C \equiv 0$.

8.1 Solution of the steady state MFE problem

We solve the linear system arising from MFE discretization of the realistic test case described in [18]. The 3D domain is subdivided into 9 zones, each of them

characterized by a different value of the hydraulic conductivity tensor whose norm varies by six order of magnitudes from 8.64×10^{-7} to $8.64 \times 10^{-1} \text{ m} \cdot \text{s}^{-1}$. This results in a very ill-conditioned steady-state problem. The problem has $N = 253216$ and a number of nonzeros $\text{nnz} = 1336168$.

Regarding the spectral properties of the matrices involved, the MFE discretization provides a very different situation, as compared to that of the Consolidation problem. Here (1,1) block K is well-conditioned, being a mass matrix whose condition number does not grow with the mesh size h . On the contrary the Schur complement matrix S can be shown to be ill-conditioned with its condition number growing as h^{-2} . The effort of the Relaxed MCP should therefore be put in properly preconditioning S . To solve our problem we chose the following parameters:

- (1) IC preconditioner for K : $\tau_K = 0.1, \text{lfil}_K = 4$;
- (2) AINV preconditioner for K : $\tau_Z = 0.5$;
- (3) IC preconditioner for S : $\tau_S = 10^{-4}, \text{lfil}_S = 50$.

Not that for this problem a simple IC(0) preconditioner for S is not sufficient to guarantee convergence of the iterative method. These parameters yields the following density values for the preconditioners: $\rho_K = 0.45, \rho_S = 2.79$.

Table 8

Iteration number and CPU times for MCP and RMCP with experimentally computed optimal value of ω for previously defined values of the tolerance tol .

$\log_{10}(\text{tol})$	ω	iter	T_{eig}	T_{prec}	T_{sol}	T_{tot}
-12	1	746	-	3.91	80.84	84.75
-10	1	589	-	3.88	63.15	67.13
-12	0.025	443	1.14	3.91	50.19	55.24
-10	0.025	259	1.18	3.88	32.45	37.51

Table 8.1 summarizes the timing and iterations results of RMCP with $\omega = 1$ using two different values of the tolerance $\text{tol} = 10^{-12}$ and $\text{tol} = 10^{-10}$. Note that, due to the ill-conditioning of the saddle point matrix, the final relative error was of the order of 10^{-8} and 10^{-6} , respectively. In the same table we report the results using RMCP with the experimentally computed value of $\omega = \frac{\beta_K}{\beta_S}$, where the leading eigenvalues were approximated using ten iterations of the DACG method. From Table 8.1 we once again appreciate the improvement in terms of iteration number and CPU time provided by RMCP with optimal ω . The elapsed time is reduced by a factor 1.5 ($\text{tol} = 10^{-12}$) or 1.8 ($\text{tol} = 10^{-10}$) with respect to the MCP.

Finally, in Figure 6 we plot the number of iterations vs ω for the two values of tol . It is shown that the experimental value of $\omega (= 0.025)$ is very close to

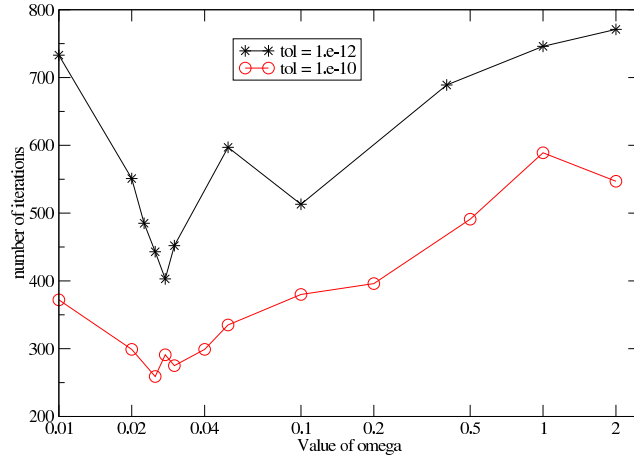


Fig. 6. Number of iterations vs ω .

the minimum of both graphs. Moreover the improvement of RMCP regarding iteration number seems to be not very sensitive to the value of ω , which therefore does not need to be assessed with high accuracy.

9 Conclusion and Future Perspectives

We have presented a class of enhanced block preconditioners for saddle point linear systems arising from coupled consolidation problems. These novel preconditioners accelerate the Mixed Constraint Preconditioners on the basis of a relaxation parameter ω which can be easily assessed by a few iterations of iterative eigensolvers to approximate the largest eigenvalue of two suitable SPD matrices. Results in the solution of small to very large coupled consolidation problems reveal that the acceleration is always effective and it reduces the number of iterations and total CPU time by a factor three in the most ill-conditioned test case.

Our approach can be applied to a wide range of problems which give rise to symmetric saddle point matrices. As a further example, we show that the RMCP proves an efficient preconditioner in the solution of a difficult steady-state problem arising from Mixed Finite Element discretization of the Darcy's law in porous media.

We expect that our approach can be efficiently employed also in the solution of the steady state Stokes problem as well as the linearized system arising from the interior point method applied to large quadratic constrained optimization problem. Future work is aimed at verifying the effectiveness of the proposed preconditioner also when addressing such problems.

Acknowledgments. We acknowledge the CINECA Iscra Award SCALPREC (2011) for the availability of HPC resources and support. We also wish to thank N. Castelletto for providing the MFE matrices and the two anonymous reviewers whose suggestions helped improve the quality of this paper.

References

- [1] D. BAÙ, M. FERRONATO, G. GAMBOLATI, AND P. TEATINI, *Basin-scale compressibility of the northern Adriatic by the radioactive marker technique*, *Gèotechnique*, 52 (2002), pp. 605–616.
- [2] M. BENZI, J. K. CULLUM, AND M. TÛMA, *Robust approximate inverse preconditioning for the conjugate gradient method*, *SIAM J. Sci. Comput.*, 22 (2000), pp. 1318–1332.
- [3] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, *Acta Numer.*, 14 (2005), pp. 1–137.
- [4] M. BENZI AND M. TÛMA, *A comparative study of sparse approximate inverse preconditioners*, *Applied Numerical Mathematics*, 30 (1999), pp. 305–340.
- [5] L. BERGAMASCHI, *Eigenvalue distribution of constraint-preconditioned symmetric saddle point matrices*, *Numer. Lin. Alg. Appl.*, (2012). Published online on October 18, 2011.
- [6] L. BERGAMASCHI, M. FERRONATO, AND G. GAMBOLATI, *Novel preconditioners for the iterative solution to FE-discretized coupled consolidation equations*, *Comp. Methods App. Mech. Engrg.*, 196 (2007), pp. 2647–2656.
- [7] ———, *Mixed constraint preconditioners for the solution to FE coupled consolidation equations*, *J. Comp. Phys.*, 227 (2008), pp. 9885–9897.
- [8] ———, *Performance and robustness of block constraint preconditioners in finite element coupled consolidation problems*, *Int. J. Numer. Methods Engrg.*, 81 (2010), pp. 381–402.
- [9] L. BERGAMASCHI, G. GAMBOLATI, AND G. PINI, *Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem*, *Numer. Lin. Alg. Appl.*, 4 (1997), pp. 69–84.
- [10] L. BERGAMASCHI, J. GONDZIO, AND G. ZILLI, *Preconditioning indefinite systems in interior point methods for optimization*, *Comput. Optim. Appl.*, 28 (2004), pp. 149–171.
- [11] L. BERGAMASCHI AND A. MARTÍNEZ, *Parallel acceleration of Krylov solvers by factorized approximate inverse preconditioners*, in *VECPAR 2004*, M. Daydè et al., ed., vol. 3402 of *Lecture Notes in Computer Sciences*, Heidelberg, 2005, Springer-Verlag, pp. 623–636.

- [12] —, *FSAI-based parallel mixed constraint preconditioners for saddle point problems arising in geomechanics*, J. Comput. Appl. Math., 236 (2011), pp. 308–318.
- [13] —, *Parallel inexact constraint preconditioners for saddle point problems*, in Euro-Par 2011, Bordeaux (France), R. N. E. Jeannot and J. Roman, eds., vol. 6853, Part II of Lecture Notes in Computer Sciences, Heidelberg, 2011, Springer, pp. 78–89.
- [14] L. BERGAMASCHI, A. MARTÍNEZ, AND G. PINI, *An efficient parallel MLPG method for poroelastic models*, CMES: Computer and Modeling in Engineering & Sciences, 49 (2009), pp. 191–216.
- [15] —, *Parallel mixed constraint preconditioners for the iterative solution of coupled consolidation problems*, in Proceedings of the Seventh Int. Conf. on Engineering Comput. Technology, Paper # 33, B. H. V. Topping, J. M. Adam, F. J. Pallares, R. Bru, and M. Romero, eds., Civil-Comp Press, 2010. CD-ROM.
- [16] L. BERGAMASCHI AND M. PUTTI, *Numerical comparison of iterative eigensolvers for large sparse symmetric matrices*, Comp. Methods App. Mech. Engrg., 191 (2002), pp. 5233–5247.
- [17] M. A. BIOT, *General theory of three-dimensional consolidation*, Journal of Applied Physics, 12 (1941), pp. 155–164.
- [18] N. CASTELLETTO, P. TEATINI, M. FERRONATO, G. GAMBOLATI, AND L. TOSI, *Mixed finite element analysis of a trial embankment at the coastland of venice, Italy*, in Computational Geomechanics, COMGEO II - Proceedings of the 2nd International Symposium on Computational Geomechanics, 2011, pp. 829–838.
- [19] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, Numer. Math., 90 (2002), pp. 665–688.
- [20] M. FERRONATO, G. GAMBOLATI, AND P. TEATINI, *Ill-conditioning of finite element poroelasticity equations*, Int. J. Solids Struct., 38 (2001), pp. 5995–6014.
- [21] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300–1317.
- [22] L. Y. KOLOTILINA, A. A. NIKISHIN, AND A. Y. YEREMIN, *Factorized sparse approximate inverse preconditionings IV. Simple approaches to rising efficiency*, Numer. Lin. Alg. Appl., 6 (1999), pp. 515–531.
- [23] L. Y. KOLOTILINA AND A. Y. YEREMIN, *Factorized sparse approximate inverse preconditionings I. Theory*, SIAM J. Matrix Anal., 14 (1993), pp. 45–58.
- [24] L. LUKŠAN AND J. VLČEK, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems*, Numer. Lin. Alg. Appl., 5 (1998), pp. 219–247.

- [25] A. MARTÍNEZ, L. BERGAMASCHI, M. CALIARI, AND M. VIANELLO, *A massively parallel exponential integrator for advection-diffusion models*, J. Comput. Appl. Math., 231 (2009), pp. 82–91.
- [26] I. PERUGIA AND V. SIMONCINI, *Block-diagonal and indefinite symmetric preconditioners for mixed finite elements formulations*, Numer. Lin. Alg. Appl., 7 (2000), pp. 585–616.
- [27] J. E. ROBERTS AND J.-M. THOMAS, *Mixed and hybrid methods*, in Handbook of numerical analysis, Vol. II, Handb. Numer. Anal., II, North-Holland, Amsterdam, 1991, pp. 523–639.
- [28] D. SILVESTER, H. ELMAN, D. KAY, AND A. WATHEN, *Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow*, J. Comput. Appl. Math., 128 (2001), pp. 261–279. Numerical analysis 2000, Vol. VII, Partial differential equations.